APPENDIX A

ABBREVIATIONS AND ACRONYMS

A.1. General.

A.1.1 Scope. This appendix contains a list of abbreviations and acronyms pertinent to MIL-STD-188-220.

A.1.2 Application. This appendix is not a mandatory part of MIL-STD-188-220. The information contained herein is intended for guidance only.

A.2. Applicable documents. This section is not applicable to this Appendix.

A.3. Abbreviations and acronyms.

(*n*)	Repeatability Factor
ABM	Asynchronous Balanced Mode
ACK	Acknowledgment
ADM	Asynchronous Disconnected Mode
ARP	Address Resolution Protocol
ASD	Adverse State Detector
ASK	Amplitude Shift Keying
BCH	Bose-Chaudhari-Hocquenghem
BER	Bit Error Rate
bps	bit(s) per second
C/R	command/response
C4I	Command, Control, Communications, Computers, and Intelligence
CCITT	International Telephone and Telegraph Consultative Committee
CDP	Conditioned Diphase
CMD	Command
CNR	Combat Net Radio
COMSEC	Communications Security
CSMA	Carrier Sensed Multiple Access
D	RE-NAD Damping coefficient
d/c	Don't Care
DAP-NAD	Deterministic Adaptive Prioritized - Network Access Delay
DARPA	Defense Advanced Research Projects Agency
dB	decibel
DC	Direct Current
DCE	Data Circuit-terminating Equipment
Dec	Decimal
DES	Destination

DIA	Unnumbered Information (PDU) With Decoupled Acknowledgement
DISC	Disconnect
DL	Data-Link Layer
DM	Disconnect Mode
DMTD	Digital Message Transfer Device
DoD	Department Of Defense
DoDISS	Department Of Defense Index Of Specifications and Standards
DPSK	Differential Phase-Shift Keying
DPTT	Delayed Push-to-Talk
DRA	Data Rate Adapter
DRNR	Decoupled Acknowledgement Receive Not Ready
DRR	Decoupled Acknowledgement Receive Ready
DTE	Data Terminal Equipment
ECP	Emergency Command Precedence
EDC	Error Detection and Correction
ETE	End-to-End
F	Final
FCS	Frame Check Sequence
FEC	Forward Error Correction
FED-STD	Federal Standard
FH	Frequency Hopping
FIPS	Federal Information Processing Standard
FPI	Field Presence Indicator
FRMR	Frame Reject
FSK	Frequency-Shift Keying
GPI	Group Presence Indicator
H-NAD	Hybrid Network Zccess Delay
HDLC	High-level Data Link Control
Hex	Hexadecimal
HF	High Frequency
HLEN	Header Length
HRT	Hop Recovery Time
Hz	Hertz
I PDU	Information PDU
IAB	Internet Architecture Board
IANA	Internet Assigned Number Authority
ICOM	Integrated COMSEC
IHL	Internet Header Length
IL	Intranet Layer
IP	Internet Protocol
ISN	Initial Sequence Number
ISO	International Organization for Standardization
JIEO	Joint Interoperability and Engineering Organization
kbps	kilobit(s) per second
KG	Key Generator

kHz	Kilohertz
KT	Keytime Delay
LOS	Line of Sight
LSB	Least Significant Bit
MI	Message Indicator
MIL-STD	Military Standard
MMTU	Minimum MTU size
MSB	Most Significant Bit
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
N(R)	Receive sequence number
N(S)	Send sequence number
NAC	Network Access Control
NAD	Network Access Delay
NATO	North Atlantic Treaty Organization
NB	Narrowband
NETCON	Network Control
NP	Network Protocol
NPDU	Network Protocol Data Unit
NRZ	Non-Return-to-Zero
NS	Number of stations
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OTAR	Over-The-Air rekeying
P	Poll
P/F	Poll/Final
P-NAD	Priority - Network Access Delay
PDU	Protocol Data Unit
PL	Physical Layer
PN	Pseudo-Noise
PSK	Phase Shift Keying
QT	Quiet Timer
R/C	Receipt/Compliance
R-NAD	Random Network Access Delay
RARP	Reverse Address Resolution Protocol
RE-NAD	Radio Embedded - Network Access Delay
REJ	Reject
REL	Relay
RF	Radio Frequency
RFC	Request For Comments
RHD	Response Hold Delay
RNR	Receive Not Ready
RR	Receive Ready
RSET	Reset
R/T	Radio/Transmitter

S/N	Signal-to-Noise ratio
S-PDU	Supervisory PDU
SABME	Set Asynchronous Balanced Mode Extended
SATCOM	Satellite Communications
SC	Single Channel
SH	Segmentation/Reassembly Header Size
SINCGARS	Single Channel Ground and Airborne Radio System
SIP	System Improvement Program
SNDCF	Subnetwork Dependent Convergence Function
SOM	Start Of Message
SOP	Start Of Packet
SP	Subscriber Precedence
SREJ	Selective Reject
ST	Satellite Time delay
STD	Standard
TBD	To Be Determined
Tc	Continuous Scheduler Interval Timer
TCP	Transmission Control Protocol
TDC	Time-Dispersive Coding
TH	Transmission Header
TIDP	Technical Interface Design Plan
TL	Traffic Load
TOS	Type Of Service
TP	Timeout Period
TRANSEC	Transmission Security
TTL	Time To Live
TWC	Transmission Word Count
U PD	Unnumbered PDU
UA	Unnumbered Acknowledgment
UDP	User Datagram Protocol
UI	Unnumbered Information
ULP	Upper Layer Protocol
URNR	Unnumbered Receive Not Ready
URR	Unnumbered Receive Ready
V(R)	Receive-state Variable
V(S)	Send-state Variable
VER	Version
VMF	Variable Message Format
WB	Wideband
XNP	Exchange Network Parameters

APPENDIX B

PROFILE

B.1. General.

B.1.1 Scope. This appendix contains the minimum set of MIL-STD-188-220 features required for joint interoperability. It is intended to guide system developers and users.

B.1.2 Application. This appendix is a mandatory part of MIL-STD-188-220. The information contained herein is intended for compliance.

B.2. Applicable documents. None.

B.3. Implementation requirements. MIL-STD-188-220 requirements are described in Section 5 and Appendices C, D, E, H, I, J and K. This appendix categorizes requirements, identified by MIL-STD-188-220 paragraph numbers, as Mandatory, Conditional or Optional. Unless otherwise specified, the category assigned to a requirement applies to all subordinate subparagraphs for the requirement. Fully compliant systems shall implement all mandatory and conditional requirements. Minimally compliant systems shall implement all mandatory requirements and some conditional requirements as described in this appendix.

B.4. Detailed Requirements.

B.4.1 Physical Layer.

B.4.1.1 Transmission Channel Interfaces. Transmission channel interfaces should be implemented as dictated by the communication device (i.e., radio) to which the system will be connected. The transmission channel interfaces for MIL-STD-188-220 compliant systems are described in 5.1.1. Minimally compliant systems shall conform to at least one of the conditional transmission channel interfaces. Fully compliant systems shall conform to all conditional requirements identified in Table B-1.

TABLE B-1. Transmission channel interfaces.

Reference	Requirement	Category
5.1.1.1	Non-return-to-zero (NRZ) interface	Conditional
5.1.1.2	Frequency-shift keying interface for voice frequency channels	Optional
5.1.1.3	Frequency-shift keying interface for single-channel radio	Optional
5.1.1.4	Conditioned diphase interface	Optional
5.1.1.5	Differential phase-shift keying interface for voice frequency channels	Optional
5.1.1.6	Packet Mode Interface	Conditional
5.1.1.7	Amplitude Shift Keying	Conditional

B.4.1.2 Physical-layer protocol. The requirements for the physical layer protocol of MIL-STD-188-220 compliant systems are described in 5.2. Fully compliant systems shall conform to all mandatory and conditional requirements identified in Table B-2.

TABLE B-2. Physical layer protocol.

Reference	Requirement	Category
5.2.1	Physical-layer protocol data unit	Mandatory
5.2.1.1	Communications security preamble and postamble	Conditional
5.2.1.2	Phasing	Conditional
5.2.1.3	Transmission synchronization field	Conditional
5.2.1.3.1	Asynchronous mode transmission synchronization field	Conditional
5.2.1.3.1.1	Frame synchronization subfield	Conditional
5.2.1.3.1.2	Robust frame format subfield	Optional
5.2.1.3.1.3	Message indicator	Conditional
5.2.1.3.1.4	Transmission wordcount	Conditional
5.2.1.3.2	Synchronous mode transmission synchronization field	Conditional
5.2.1.3.2.1	Frame synchronization subfield	Conditional
5.2.1.3.2.2	Robust frame format subfield	Optional
5.2.1.3.2.3	Message indicator	Conditional
5.2.1.3.2.4	Transmission wordcount	Conditional
5.2.1.3.3	Packet mode transmission synchronization field	Conditional
5.2.1.3.4	Multi-dwell protocol	Optional
5.2.1.4	Data field	Mandatory
5.2.1.5	Bit synchronization field	Conditional
5.2.2	Net access control related indications	Mandatory
5.2.3	Physical-layer to upper-layer interactions	Mandatory

Minimally compliant systems shall implement all mandatory physical layer protocol requirements. Also:

a) minimally compliant C^4I systems using the NRZ interface shall implement all conditional requirements for the synchronous mode transmission field (and conditional subparagraphs), communications security preamble and postamble and phasing.

b) minimally compliant C^4I systems using the Packet Mode interface shall implement all conditional requirements for the packet mode transmission field.

c) minimally compliant C^4I systems using the ASK interface shall implement all conditional requirements for the asynchronous mode transmission field (and conditional subparagraphs), communications security preamble and postamble and phasing.

B.4.2 Data-link layer. The requirements for the data layer protocol of MIL-STD-188-220 compliant systems are described in 5.3. Fully compliant system shall conform to all mandatory and conditional requirements identified in Table B-3.

Minimally compliant systems shall implement all mandatory data link layer requirements. Also:

a) minimally compliant C^4I systems using the synchronous mode transmission synchronization field shall implement all conditional requirements for time dispersive coding and error detection and correction (including subparagraphs).

b) minimally compliant C^4I systems using the packet mode transmission synchronization field shall implement all conditional requirements for the scheduler and the queue precedence and queue length fields of the transmission queue subfield.

c) minimally compliant C^4I systems using the asynchronous mode transmission synchronization field shall implement all conditional requirements for time dispersive coding and error detection and correction (including subparagraphs).

TABLE B-3. Data link layer.

Reference	Requirement	Category
5.3.1	Transmission Header	Mandatory
5.3.1.1	Selection bits	Mandatory
5.3.1.2	Topology Update Identifier	Mandatory
5.3.1.3	Transmission queue subfield	Mandatory
5.3.1.3.1	T-bits	Mandatory
5.3.1.3.2	Queue precedence	Conditional
5.3.1.3.3	Queue length	Conditional
5.3.1.3.4	Data link precedence	Mandatory
5.3.1.3.5	First subscriber number	Mandatory
5.3.2	Net access control	Mandatory
5.3.2.1	Scheduler	Conditional
5.3.3	Types of procedures	Mandatory
5.3.3.1	Type 1 operation	Mandatory
5.3.3.2	Type 2 operation	Optional
5.3.3.3	Type 3 operation	Mandatory
5.3.3.4	Type 4 operation	Optional
5.3.3.5	Station Class	---
5.3.4	Data-link frame	Mandatory
5.3.4.1	Types of frames	Mandatory
5.3.4.1.1	Unnumbered frame	Mandatory
5.3.4.1.2	Information frame	Optional
5.3.4.1.3	Supervisory frame	Optional
5.3.4.2	Data-link frame structure	Mandatory
5.3.4.2.1	Flag sequence	Mandatory
5.3.4.2.2	Address fields	Mandatory
5.3.4.2.3	Control field	Mandatory
5.3.4.2.3.1	Type 1 operations	Mandatory
5.3.4.2.3.2	Type 2 operations	Optional
5.3.4.2.3.3	Type 4 operations	Optional
5.3.4.2.3.4	Poll/final bit	Mandatory
5.3.4.2.3.5	Sequence numbers	Optional
5.3.4.2.3.6	Identification numbers	Optional
5.3.4.2.3.7	Precedence	Optional
5.3.4.3	Data-link PDU construction	Mandatory
5.3.5	Operational parameters	Mandatory
5.3.5.1	Type 1 operational parameters	Mandatory
5.3.5.2	Type 2 operational parameters	Optional
5.3.5.3	Type 4 operational parameters	Optional

TABLE B-3. Data link layer. - Continued.

5.3.6	Commands and responses	Mandatory
5.3.6.1	Type 1 operation commands and responses	Mandatory
5.3.6.2	Type 2 operation commands and responses	Optional
5.3.6.3	Type 4 operation commands and responses	Optional
5.3.7	Description of procedures by type	Mandatory
5.3.7.1	Description of Type 1 procedures	Mandatory
5.3.7.2	Description of Type 2 procedures	Optional
5.3.7.3	Description of Type 4 procedures	Optional
5.3.8	Data-link initialization	Mandatory
5.3.8.1	List of data-link parameters	Mandatory
5.3.8.1.1	Type 1 logical data link parameters	Mandatory
5.3.8.1.2	Type 2 data link parameters	Optional
5.3.8.1.3	Type 4 data link parameters	Optional
5.3.9	Frame transfer	Mandatory
5.3.9.1	PDU transmission	Mandatory
5.3.9.2	Data-link concatenation	Mandatory
5.3.9.3	Physical-layer concatenation	Optional
5.3.9.4	PDU transmissions	Mandatory
5.3.10	Flow control	Mandatory
5.3.10.1	Type 1 flow control	Mandatory
5.3.10.2	Type 2 flow control	Optional
5.3.10.3	Type 4 flow control	Optional
5.3.11	Acknowledgment and response	Mandatory
5.3.11.1	Acknowledgment	Mandatory
5.3.11.1.1	Type 1 Acknowledgment	Mandatory
5.3.11.1.2	Type 2 Acknowledgment	Optional
5.3.11.1.3	Type 4 Acknowledgment	Mandatory
5.3.11.2	Quiet mode	Mandatory
5.3.11.3	Immediate retransmission	Optional
5.3.12	Invalid frame	Mandatory
5.3.13	Retransmission	Mandatory
5.3.14	Error detection and correction	Conditional
5.3.15	Data scrambling	Optional
5.3.16	Link layer interactions	Mandatory

B.4.3 Intranet protocol. The requirements for the Intranet protocol of MIL-STD-188-220 compliant systems are described in 5.4. Fully compliant systems shall conform to all mandatory requirements identified in Table B-4.

TABLE B-4. Intranet protocol.

Reference	Requirement	Category
5.4.1.1	Intranet header	Mandatory
5.4.1.2	Topology update	Optional
5.4.1.3	Topology update request message	Optional
5.4.1.4	Intranet layer interactions	Mandatory
5.4.2	Subnetwork Dependent Convergence Function (SNDCF)	Mandatory
5.4.2.1	Determine destination function	Mandatory
5.4.2.2	Address mapping function	Mandatory
5.4.2.3	Type of service function	Mandatory
5.4.2.4	Intranet send request	Mandatory

B.4.4. Network Access Control Algorithm. The requirements for network access control implementation in MIL-STD-188-220 compliant systems are described in Appendix C. Fully compliant system shall conform to all mandatory requirements identified in Table B-5. Requirements for RE-NAD are optional because the algorithm has not yet stabilized.

TABLE B-5. Network access control.

Reference	Requirement	Category
C3.	Network timing model	Mandatory
C3.1	Network timing model definitions	Mandatory
C3.2	Network timing model parameters	Mandatory
C4.	Network access control	Mandatory
C4.1	Network busy sensing function	Mandatory
C4.1.1	Data network busy sensing	Mandatory
C4.1.2	Voice network busy sensing	Mandatory
C4.1.3	Net busy detect time	Mandatory
C4.2	Response hold delay	Mandatory
C4.3	Timeout period	Mandatory
C4.4	Net access delay	Mandatory
C4.4.1	Random network access delay	Mandatory
C4.4.2	Prioritized network access delay	Mandatory
C4.4.3	Hybrid network access delay	Mandatory
C4.4.4	Radio embedded network access delay (RE-NAD)	Optional
C4.4.5	Deterministic Adaptable Priority-Net Access Delay	Mandatory
C4.5	Voice/data network sharing	Mandatory

Minimally compliant systems shall implement all mandatory network access control requirements. Also minimally compliant C^4I systems using the packet mode interface shall implement all conditional requirements for radio embedded network access delay.

B.4.5 Communications Security Standards. The communications security requirements for MIL-STD-188-220 compliant systems are described in Appendix D. There are no communications security requirements for systems using the packet mode transmission synchronization field. Fully compliant systems shall implement all conditional requirements identified in Table B-6.

Minimally compliant systems using the synchronous mode or asynchronous mode transmission synchronization field shall implement the conditional requirements for the traditional COMSEC transmission frame.

TABLE B-6. Communications security standards.

Reference	Requirement	Category
D5.	Detailed requirements	Conditional
D5.1	Traditional COMSEC transmission frame	Conditional
D5.2	Embedded COMSEC transmission frame	Optional

B.4.6. CNR Management Processes. CNR management process requirements are defined in Appendix E. All CNR management process requirements are optional.

B.4.7. Intranet Topology Update. Intranet Topology Update requirements are described in Appendix H. All systems implementing the optional Topology Update and Topology Update Request message requirements described in 5.4 also shall implement all requirements in Appendix H.

B.4.8. Source Directed Relay. The Intranet relay requirements for MIL-STD-188-220 compliant systems are described in Appendix I. All systems shall implement all requirements in paragraph I4 (and its subparagraphs).

B.4.9. Robust Communications Protocol. Robust communications protocol requirements are described in Appendix J. All systems implementing the optional robust frame format subfield described in 5.2.1.3.1.2 or 5.2.1.3.2.2 also shall implement all requirements in Appendix J.

B.4.10. Bose-Chaudhari-Hocquenghem (15, 7) Coding Algorithm. The Bose-Chaudhari-Hocquenghem (15, 7) Coding Algorithm is described in Appendix I. All systems implementing the optional Robust Communications Protocol described in Appendix J also shall implement all requirements in Appendix K.

APPENDIX C

NETWORK ACCESS CONTROL ALGORITHM

C.1. General.

C.1.1. Scope. This appendix describes the network access control (NAC) algorithm to be used in the DMTD and interfacing C.4I systems.

C.1.2. Application. This appendix is a mandatory part of MIL-STD-188-220. The information contained herein is intended for compliance.

C.2. Applicable documents. This section is not applicable to this appendix.

C.3 Network Timing Model. The network access control protocol shall be used to detect the presence of active transmissions on a multiple-subscriber-access communications network and shall provide a means to preclude data transmissions from conflicting on the network. The network access control protocol is based on a generic network timing model. All stations on a network shall use the same network access control protocol and timing parameter values in order to maintain network discipline.

C.3.1 Network Timing Model Definitions. A network station consists of a DCE and a DTE. The DTE is the data device that performs the MIL-STD-188-220 protocol. The DCE includes all equipment external to the DTE (e.g., a radio with or without external COMSEC) that is used to provide a communications channel for the DTE. The interface between the DTE and DCE can operate in synchronous, asynchronous, or packet mode. The interface is synchronous if the DCE provides all required clocks to the DTE. The packet mode interface is a synchronous interface that conforms to CCITT X.21. For synchronous mode, the bit rate (n) is the rate of the transmit clock supplied by the DCE. If the DCE does not provide clocks to the DTE, the interface is asynchronous. For asynchronous mode, the bit rate (n) is the rate at which the DTE transmits. The data link bit rate is defined as the effective bit rate at which the physical layer transmits the data bits. The data link bit rate is usually the same as the bit rate (n) at the physical layer, except for the PSK/DPSK modems (refer to MIL-STD-188-110). The robust protocol case is separately described in Appendix J.

C.3.2 Network Timing Model Parameters. The parameters of the network timing model are general enough to model interactions with a wide variety of DCEs. All parameters are specified at the DTE to DCE interface and are in units of milliseconds with a resolution of one millisecond. Parameters may have a value of zero if they are not applicable to the DCE being used. Network timing model parameters are shown in Figure C-1.

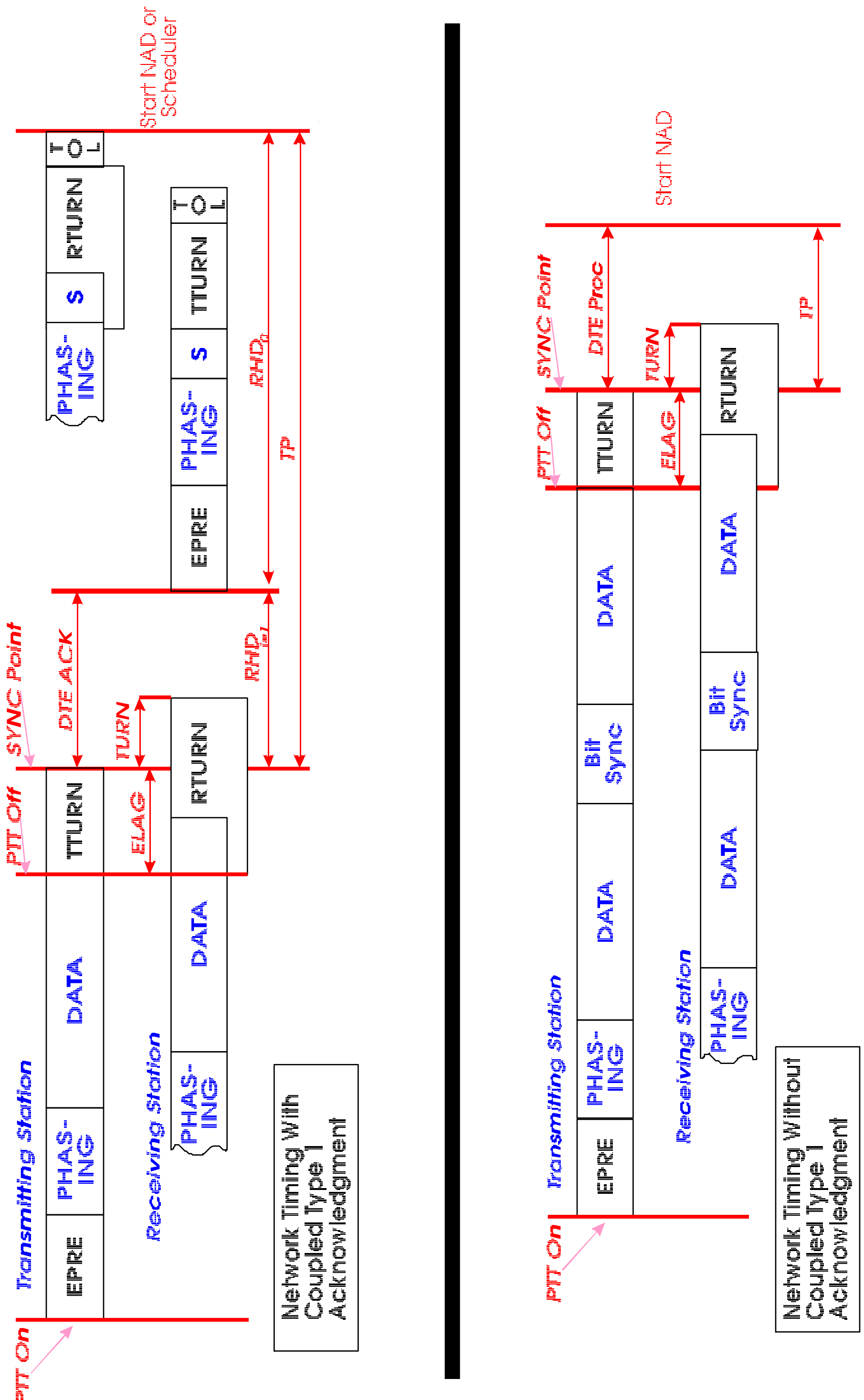


FIGURE C-1. Network timing model

C.3.2.1 Equipment Preamble Time (EPRE). EPRE is the time from when the DTE initiates a transmission, often by asserting Push-to-Talk (PTT), until the transmitting DTE sends to its DCE the first bit of information that will be delivered to the receiving DTE. EPRE is a characteristic of the DCE. It accounts for DCE start up time, including time required for radio power up and transmission of COMSEC and other DCE preambles. EPRE can have a value between 0 and 30,000 milliseconds.

a. For Synchronous mode, EPRE is measured from PTT until the DCE provides a clock to the DTE for its first bit of information. For the purposes of the Network Timing Model, it is assumed that the DTE will begin sending information to the DCE with the first clock edge provided by the DCE. During this time, the DTE sends nothing.

b. For Asynchronous mode, EPRE is measured from PTT until the first signal transmitted by the sending DTE is also transmitted by the sending DCE to receiving DCEs. This accounts for time that the transmitting DCE is not listening to signals sent by the transmitting DTE. During this time, the transmitting DTE may send an alternating sequence of one and zero bits.

c. For Packet mode, EPRE is measured from the time the DTE indicates it is ready to transmit (by asserting the C-lead and transmitting flags on the T-lead as described in 6.3.3.1.2) until the DCE indicates it is ready to accept information from the DTE (by transmitting flags to the DTE on the R-lead as described in 6.3.3.1.2).

C.3.2.2 Phasing Transmission Time (PHASING). PHASING is the time the DTE must send an alternating sequence of one and zero bits after the completion of EPRE and prior to sending the first bit of DATA. PHASING can be needed due to characteristics of the DCE, DTE, or both. PHASING can have a value between 0 and 10,000 milliseconds. The DTE shall use the DCE bit rate to compute the number of PHASING bits to transmit.

a. For Synchronous mode, PHASING can be needed if the DCE delivers extraneous clock edges to the DTE prior to the start of a valid, continuous transmit clock or if the DCE provides a transmit clock to the DTE before it is ready to reliably deliver bits from the DTE to receiving DCEs.

b. For Asynchronous mode, PHASING is often needed by the receiving DTE to achieve bit synchronization.

c. For Packet mode, PHASING is always zero.

C.3.2.3 Data Transmission Time (DATA). DATA is the time during which the transmitting DTE sends transmit data to its DCE. Transmit data includes all fields shown in Figure 5. This includes embedded COMSEC information shown in Figure 5b. It also includes transmission of concatenated frames (including bit synchronization between physically concatenated frames) as shown in Figure 3. DATA shall begin immediately after the end of PHASING. The transmitting DTE shall indicate end of transmission immediately after the last bit of data is sent to the DCE.

C.3.2.4 Coupled Acknowledgment Transmission Time (S). S is a special case of DATA, where the Data Field shown in Figure 5 contains only one Type 1 URR, URNR or TEST response frame with the F-bit set and no information field. For these frames, the length of the fields in Figure 5 (including zero bit insertion) used in network timing equations when the Multi-Dwell protocol and convolutional coding are not used shall be:

a. the 64-bit message synchronization field,

b. an optional embedded COMSEC MI field,

c. the 168-bit Transmission Wordcount and Transmission Header TDC block, and

d. 80 bits if neither the FEC nor TDC function is selected, 168 bits if only FEC is selected, and 384 bits if both FEC and TDC are selected.

The sum of these components is transmitted at the data link bit rate.

C.3.2.5 Equipment Lag Time (ELAG). ELAG is the time from when the last bit of DATA is sent by the transmitting DTE until the time when the same last bit of DATA is delivered to the receiving DTE by the receiving DCE. ELAG is a characteristic of the DCEs. It accounts for frequency hopping throughput delays, satellite transmission delays, Packet Mode radio-embedded FEC delays and other related delays. The end of ELAG is the synchronization point for the Timeout Period (TP) and Response Hold Delay (RHD) Timers.

C.3.2.6 Turnaround Time (TURN). TURN is the time from the end of ELAG until the end of TTURN or RTURN, whichever is later. TURN is computed using the equation:

$$\text{TURN} = \text{Maximum}((\text{TTURN} - \text{ELAG}), (\text{RTURN} - \text{ELAG}))$$

where:

a. TTURN is the time from when the transmitting DTE indicates end of transmission at the end of DATA until the transmitting DCE is ready to begin a new transmit or receive operation. TTURN is a characteristic of the DCE. It includes time when the transmitting DCE sends COMSEC or other postambles after transmitting all DATA.

b. RTURN is the time from when the transmitting DTE indicates end of transmission at the end of DATA until the receiving DCE is ready to begin a new transmit or receive operation. RTURN is a characteristic of the DCE.

c. ELAG may be either larger or smaller than TTURN, but is always less than or equal to RTURN, as shown in Figure C-2.

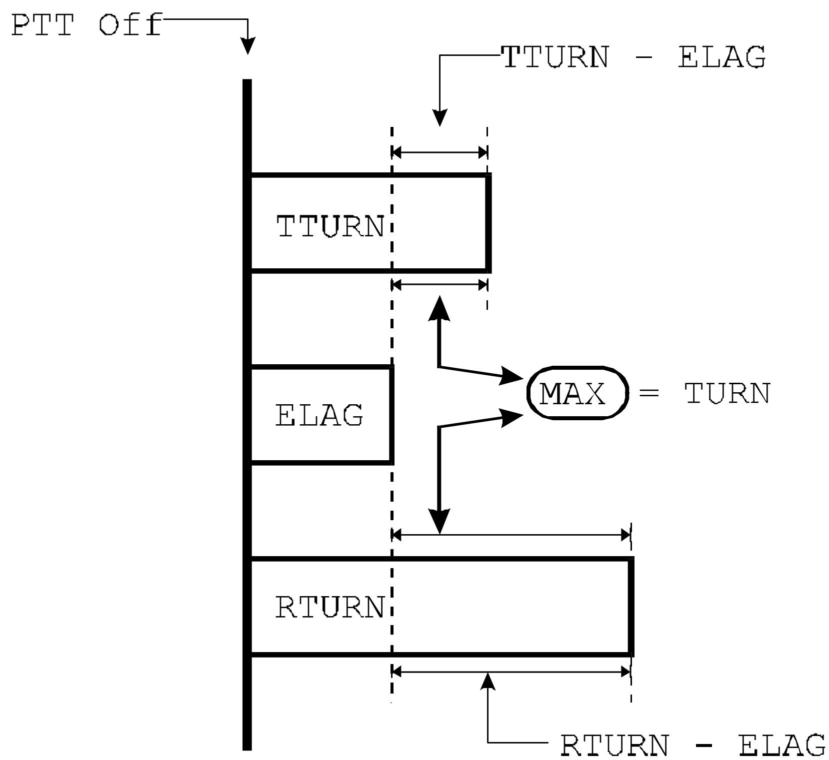


FIGURE C-2. Turnaround time (TURN) calculation

C.3.2.7 DTE Ack Preparation Time (DTEACK). DTEACK is the time from the end of ELAG until the slowest DTE on the network can process any possible Type 1 frame requiring a coupled acknowledgment, prepare the coupled Type 1 acknowledgment frame, and begin sending its coupled acknowledgment frame to its DCE. DTEACK is a characteristic of the DTE. Unless a larger value is known, use the value TURN for the particular radio and operating environment as the default value for DTEACK.

C.3.2.8 DTE Processing Time (DTEPROC). DTEPROC is the time from the end of ELAG until the slowest DTE on the network can begin sending its next transmission to its DCE after receiving DATA not requiring a coupled, Type 1 acknowledgment. DTEPROC is a characteristic of the DTE. Unless a larger value is known, use the value TURN for the particular radio and operating environment as the default value for DTEPROC.

C.3.2.9 DTE Turnaround Time (DTETURN). DTETURN is the time required for the DTE to begin a transmit operation when starting in a listening for receive state. DTETURN is the time required for the DTE to stop listening for received data or squelch and to start transmitting (including time for transmit relays for PTT to close). DTETURN shall have a fixed value of 10 milliseconds.

C.3.2.10 Tolerance Time (TOL). TOL is a time value used to compensate for variances in the time needed to transmit a coupled Type 1 acknowledgment frame. TOL shall not exceed 500 milliseconds. TOL shall be selectable with 50 milliseconds as its default value.

C.4. Network access control. The stations shall implement the following four basic NAC subfunctions:

a. network busy sensing

b response hold delay (RHD)

c. timeout period (TP)

d. network access delay (NAD)

C.4.1. Network busy sensing function. The network busy function is used to establish the presence of a data or voice signal at the receiving station due to activity on the net. Network busy sensing for a data signal shall be provided. Network busy sensing for a voice signal may be provided.

C.4.1.1. Data network busy sensing. When receiving a data transmission, network busy shall be detected within a fixed time. Parameter B shall be used to compute this fixed time. For synchronous mode B shall be less than or equal to (32/n) seconds. For asynchronous mode B shall be less than or equal to (64/n) seconds. For packet mode B shall be less than or equal to 250 milliseconds. Upon detection of data network busy, the data-link network busy indicator shall be set. Setting the data-link network busy indicator shall inhibit all message transmissions, including coupled response messages. The data-link network busy indicator shall be reset upon indication from the physical layer that neither voice nor digital data is being detected by the station.

C.4.1.2. Voice network busy sensing. Network busy due to a voice transmission may be detected. If voice transmissions are not detected, this function shall report that the network is never busy due to a voice transmission. Upon detection of voice network busy, the data-link network busy indicator shall be set. Setting the data-link network busy indicator shall inhibit all message transmissions, including coupled response messages. The data-link network busy indicator shall be reset upon indication from the physical layer that neither voice nor digital data is being detected by the station.

C.4.1.3. Network busy detect time. The time allowed to detect data network busy shall be the same for all stations on the network. This Net_Busy_Detect_Time is a key factor in achieving both throughput and speed of service. Where a communication media provides capabilities to detect data network busy more quickly than given by the formula below, the use of these capabilities is strongly encouraged. In these cases, Net_Busy_Detect_Time can be set to reflect the capabilities of the media. Where the communication media does not provide special capabilities or these capabilities cannot be used by all stations on the network, the station shall examine received data to detect data network busy. In these cases, the time allowed to detect data network busy shall be given by the formula:

Net_Busy_Detect_Time = EPRE + ELAG + B

NOTE: Parameters EPRE and ELAG are initialized locally or learned using the XNP messages described in Appendix E. Net_Busy_Detect_Time can also be learned using the XNP messages described in Appendix E.

C.4.2. Response hold delay. An RHD_0 period and an individual RHD value are calculated to determine the time that an addressed receiving station delays before sending a Type 1 response PDU upon receiving a Type 1 command PDU (UI and TEST) requesting acknowledgment (that is, P-bit set to 1 and addressed to the station's individual or special address). The RHD controls network access and the NAD algorithm is suspended during this period. An RHD_0 period is the worst-case amount of time that a single response takes. The individual RHD is the time at which a particular station waits before accessing the network. If the scheduler is running, immediate scheduling should be used for Type 1 Acknowledgment. The individual RHD value to be used shall be determined by the position of the receiving station's individual or special address in the PDU destination portion of the address field. The Reserved Address (0)in the destination portion of the address field shall be ignored. That is, when calculating an individual RHD value, the Reserved Address shall not be considered to occupy a position in the destination portion of the address field. The calculated values for RHD_i, TP, and NAD are computed to the nearest millisecond. The RHD time shall start precisely at the end of ELAG. All stations on a subnetwork shall use the same values in calculating RHD. These values can be initialized locally or learned, using the XNP messages described in Appendix E.

a. The RHD_0 period shall be calculated by the following formula:

$$RHD_O = EPRE + PHASING + S + ELAG + TURN + TOL$$

b. The TP shall be calculated by all stations on the network/link as follows:

$$TP = (j * RHD_0) + TOL + TURN$$

where j = The total number of destination link addresses - to include special and individual but not group or global addresses - for this transmitted frame. The transmitting station shall not include special address 3 in the total for j, and the value of all non-integer variables (that is, RHD_0, TOL, and TURN) in the TP equation are rounded to the nearest one thousandth.

c. The individual addressed station's response hold delay (RHDi) shall be calculated by

$$RHD_i = (i - 1) * RHD_0 + \text{Maximum}(DTEACK, TURN)$$

The variable i (where $1 \leq i \leq 16$) is the individual station's position in the destination portion of the address field.

C.4.3 Timeout period. TP is the time all stations shall wait before they can schedule the NAD. During this window of time, the transmitting station shall wait to receive the anticipated Type 1 coupled acknowledgment response frame(s), if any, from all applicable addressed stations. The parameter values used to compute TP shall be the same for all stations on a subnet unless immediate retransmission has been selected. When immediate retransmission has been requested, the sending station shall compute the timeout period using only individual addresses and special addresses 1 and 2. All receiving stations shall compute the timeout period using the individual addresses and special addresses 1, 2 and 3. The calculated value of TP is computed to the nearest millisecond. The TP time shall start precisely at the end of ELAG. A retransmission of a Type 1 P-bit frame shall be executed whenever TP has been exceeded without expected acknowledgments having been received from all Type 1 individual and special destinations. Prior to retransmission, the address field of the frame shall be modified to delete the destination station(s) that previously acknowledged the frame. Operationally, TP shall be used as follows:

a. Upon termination of a message transmission that requires an immediate response, the transmitting station shall set the TP timer. If the transmitting station does not receive all the expected responses (TEST, URR, or URNR) within the TP, and if the number of transmissions is less than the Maximum Number of Transmissions data link parameter, the station shall retransmit the frame when it is the highest precedence frame to send. For all stations, if a Type 1 (P-bit=0), Type 2 or Type 4 frame is received when a response-type frame is expected, the newly received frame shall be processed. The RHD and TP timers shall not be suspended and the TP procedures in use for the Type 1 (P-bit=1) frame shall be continued. Response procedures, if any, for the newly received frame shall commence after the conclusion of the ongoing TP procedures. If the unexpected frame is a Type 1 (P-bit=1) frame the current TP procedure is aborted and the newly received Type 1 (P-bit=1) TP procedure shall be started.

b. After a station transmits or receives data that does not require a Type 1 coupled acknowledgment, and is not itself a Type 1 coupled acknowledgment, all stations except those using RE-NAD shall compute TP as:

$$TP = \text{Maximum}(DTEPROC, TURN)$$

c. Upon receiving a Type 1 coupled acknowledgment, a station shall determine whether it thinks a timeout period is already in progress. If no timeout period is in progress, the station shall compute TP using the following equation and shall start a timeout period precisely at the time the last bit of data for the Type 1 coupled acknowledgment was received.

$$TP = (15 * RHD_0) + TOL + TURN$$

NOTE: RHD_0 is as defined in 4.2.

C.4.4 Network access delay. NAD is defined as the time a station with a message to send shall wait to send a frame after the TP timer has expired. NAD discipline is based on an infinite sequence of "slots" that begin when the TP timer has expired. Slots are defined to be long enough so that all stations on the network will detect a station transmitting at the beginning of a slot prior to the beginning of the next slot. The duration of each slot is NET_BUSY_DETECT_TIME. All transmissions, except the coupled acknowledgments, shall begin at the start of the next NAD slot.

There are five schemes for calculating NAD. The five schemes are defined below. Four schemes (R-NAD, P-NAD, H-NAD AND DAP-NAD) compute a value F for each station on the net. The F value is the number of NAD slots that each station will wait before transmitting, if it has any information to send.

The random network access delay (R-NAD) scheme provides all stations with an equal chance to access the network. The prioritized network access delay (P-NAD) scheme ensures the highest precedence station with the highest priority message will access the network first. In the case of RE-NAD, network access delay is computed by the radio. With RE-NAD the DTE (DMTD or C.4I system) does not compute network access delay but does schedule channel access opportunities at which an access attempt can be initiated by the DTE. DAP-NAD, like P-NAD, ensures the highest priority message will access the network first. It does not ensure first access by highest precedence station however. The hybrid network access delay (H-NAD) scheme combines random access with the preferential access by frame priority. The random and hybrid schemes might result in a collision (the same NAD value for two stations). The P-NAD and DAP-NAD schemes always produce a unique NAD value for each station. In all of the NAD schemes, if the TP timer is active, the stations with frames to transmit shall wait for the TP timer to expire before the NAD is started. If the TP timer is not active, the station shall calculate its NAD using the proper NAD scheme for the network. Each NAD scheme produces a set of allowed access periods. The network may be accessed only at the beginning of one of those periods. If a station using P-NAD, DAP-NAD or H-NAD is waiting for its NAD time and a higher priority frame becomes available for transmission, the station may shorten its NAD time to a time it would have computed if it had computed its original NAD time using the new, higher frame priority. Below are the frame reception and transmission procedures:

a. A station shall analyze a received frame to determine if a TP timer must be set. After the frame check sequence has been verified, the address and control fields are analyzed. If the received frame is either a UI or TEST frame and the poll bit is set to 1, then a TP timer is set. Any other pending frames for transmission shall be placed on hold. If the received frame was not a UI or TEST frame with the poll bit set, a NAD value shall be computed and initiated after the TP timer expires. An R-NAD or H-NAD value shall be calculated and initiated if the network busy status is clear. DAP-NAD values need to be recalculated after each transmission. The calculated value of NAD is rounded to the nearest millisecond.

b. If a station does not have a frame to transmit, it shall compute a NAD time using routine priority for its calculations. If the NAD time arrives before a frame becomes available to transmit or frame(s) are not yet encoded for transmission, the station shall compute and use a new NAD time. The starting time for the new NAD shall be the same as the starting time for the NAD that was just completed. The F value used in computing the NAD shall be the sum of the F value used in the NAD just completed, plus a value dependent on the NAD in effect.

1) For P-NAD this value shall be (NS + 1). This creates another group of NAD slots for all stations on the network. Adding this value at all stations preserves the algorithmic collision prevention property of P-NAD.
2) For R-NAD this value shall be [(3/4) * NS + 1]. Adding the same constant value at all stations preserves the random property of R-NAD.
3) For H-NAD this value shall be 1 if the station has an urgent or priority frame to transmit and (Routine MAX + 1 - Routine MIN) if a station has only a routine frame(s) or no frame(s) to transmit. The value 1 preserves the intent of H-NAD that is to grant preferential network access to stations with urgent or priority frames to send. The value (Routine MAX + 1 - Routine MIN) preserves the random property of H-NAD for stations with only routine frames to send.
4) For RE-NAD, F is not used.
5) For DAP-NAD this value shall be (NS). This creates another group of NAD slots for all stations on the network. Adding this value at all stations preserves the algorithmic collision prevention property of DAP-NAD.

c. All stations on the network shall continue to sense the link for data or voice network busy and shall withhold transmission until the appropriate NAD period has expired. NAD shall be calculated using the formula:

NAD = F * Net_Busy_Detect_Time + Max(0, F-1) * DTETURN

where Net_Busy_Detect_Time is as defined in C.4.1.3 and DTETURN is as defined in C.3.2i.

C.4.4.1 Random network access delay. The R-NAD calculation method shall ensure that each station has an equal chance of accessing the network. The random nature also may provide a resolution if an access conflict occurs. Each attempt to access the network potentially can use a

NAD value different from the station's previous value. The integer value of F shall be obtained from a pseudorandom number generator. The range of the pseudorandom number depends on the number of stations (NS) in the network. F shall be an integer value (truncated) in a range between 0 and (3/4)NS. NS can be learned through the XNP join exchange, or fixed by a system parameter established at initialization.

C.4.4.2 Prioritized network access delay. The P-NAD calculation method shall ensure that the network access precedence order assigned to subscribers is preserved. Each station shall calculate three unique P-NAD values, one for each of the three frame precedence levels. The integer value of F shall be calculated as:

$$F = SP + MP + IS$$

where:

SP = the station priority:
- SP = (subscriber rank -1) for the initial transmission; and
- SP = 0 for subsequent transmissions.

MP = the message precedence:
- MP = 0 for all urgent messages;
- MP = (NS + 1) for all priority messages;
- MP = 2 * (NS + 1) for all routine messages,
- where NS is the number of subscribers on the network.

IS = the initial/subsequent factor:
- IS = 0 for the initial transmission, and
- IS = NS for subsequent transmissions.

Only one station on the network uses the subsequent factor. That is the station that transmitted last on the net. However, transmissions of coupled Type 1 acknowledgments do not count as transmissions for the purpose of determining which station transmitted last.

C.4.4.3 Hybrid network access delay. The H-NAD calculation method ensures that network access delay times are shorter for higher priority frames, while maintaining equal access chances for all stations. Each priority level has a distinct range of pseudorandom F values determined by the number of stations in the subnetwork, the network percentage of the particular priority level frames, and the traffic load. The integer value of F shall be calculated as

$$F = MIN + RAND * (MAX - MIN)$$

where:

RAND = pseudorandom number in the range 0.0 to 1.0

MAX and MIN are integer values defining the ranges:

Urgent_MIN = 0, for urgent frames

Urgent_MAX = USIZE + 1, for urgent frames

Priority_MIN = Urgent_MAX + 1, for priority frames

Priority_MAX = Priority_MIN + PSIZE + 1, for priority frames

Routine_MIN = Priority_MAX + 1, for routine frames

Routine_MAX = Routine_MIN + RSIZE + 1, for routine frames

USIZE = the additional number of random numbers generated for urgent frames

PSIZE = the additional number of random numbers generated for priority frames

RSIZE = the additional number of random numbers generated for routine frames

where the minimum MIN/MAX range size is 2.

The additional range sizes (xSIZE) are integers based on the percent of frames expected at a specific priority level (%priority_level) and the number of stations adjusted (ADJ_NS) by the expected traffic load (TL). NS, %priority_level, and TL, may be input using the XNP messages or by initialization input. xSIZE is rounded to the nearest non-negative integer.

USIZE = %U * ADJ_NS, %U = percentage of urgent frames (default 25%)

PSIZE = %P * ADJ_NS, %P = percentage of priority frames (default 25%)

RSIZE = %R * ADJ_NS, %R = percentage of routine frames or 100% - (%U + %P)
(default 50%)

where the adjusted number of stations increases if the expected TL is heavy and decreases if the traffic load is light. The minimum random number range at each of the three priority levels is 2, so 6 stations are subtracted from the adjusted number of stations.

ADJ_NS = INTEGER (NS * TL) -6 or = 1 (whichever is greater)

where:

TL = 1.2 Heavy Traffic Load
= 1.0 Normal Traffic Load
= 0.8 Light Traffic Load

C.4.4.4. Radio embedded network access delay (RE-NAD).

C.4.4.4.1. RE-NAD media access. The radio embedded network access delay (RE-NAD) DTE data link layer uses a 1-persistent channel access protocol between the DTE (DMTD or C.4I system) and DCE. When the continuous scheduler interval timer (Tc) expires and the previous series of concatenated frames was successfully transmitted, a new series of frames is sent to the physical layer. If there is a pending series of concatenated frames, its transmission is requested again. It should be noted that the physical layer holds the series of concatenated frames when channel access has been denied. If channel access was denied a new Tc timer is calculated and channel access for transmission of the pending series of concatenated frames is requested when the new Tc timer expires. If a higher precedence individual frame becomes available for transmission, the concatenated frames shall be re-built to include the higher precedence frame.

For the Type 1 acknowledgment, the RE-NAD portions in both DTE and DCE are suspended and channel access is controlled by the RHD and TP processes. The RE-NAD algorithm is resumed following expiration of the TP timer.

C.4.4.4.1.1. Random schedule interval. In order to achieve high performance radio network communication, efficient channel access and multi-level precedence, a 1-persistence RE-NAD algorithm is implemented in the DTE. This algorithm uses the "continuous scheduler" concept to provide a random distribution of timing for channel access via the Tc interval timer. The Tc interval timer is the sum of a fixed interval and a random interval. Toffset is the fixed interval. The fixed interval is dependent on the local station's recent use of the channel and is described more fully in C.4.4.4.1.2. The random interval is dependent on network population, network connectivity, traffic load and the local station's recent use of the channel. It is discussed in C.4.4.4.1.3 and C.4.4.4.1.4.

The value of Tc is recalculated periodically from the expression:

Tc = Toffset + uniform_random(schedint)

where uniform_random (schedint) is a uniform random number function using the range 0-schedint. Uniform_random returns an integer value.

The record of transmit traffic load at the local station is updated after every transmission attempt by the local station, which modifies the value of Toffset. Thus, TC is to be updated after every local transmission attempt.

C.4.4.4.1.2. Calculation of the scheduler offset. The parameter "Toffset" in C.4.4.4.1.1 is a function of the average transmit duration. A transmission is composed of unnumbered, supervisory and information frames. Combinations of these frames are concatenated into a series of frames for transmission. Toffset is calculated as follows:

$$\text{Toffset} = 2 * \text{average transmit duration of the series of concatenated frames}$$

$$\text{Toffset} = 2 * \text{Ttrans}, \; 1.0 \le \text{Toffset} \le 10 \text{ seconds}$$

where Ttrans = Average concatenated frame transmit duration

The average transmit duration of a series of concatenated frames is determined from the knowledge of the average length of the series of frames in bits divided by the effective on the air information transfer rate. The average length of the series of concatenated frames is computed based on the length of the last four series of concatenated frames transmitted by the local station. Toffset is bounded by 1.0 to throttle the channel by not allowing a station continuous access to the channel. The maximum of 10 seconds places an upper bound on the amount of time a station must wait between channel access attempts when long messages on a low rate channel are used. Toffset is to be updated after every transmission by the local station.

If the scheduler expires and there are no PDUs to transmit, the number of bits equal to one half the effective on the air information transfer rate (bps) will be entered into the record containing the last four transmissions. The value of Ttrans will default to 0.5 second. This will allow the Toffset parameter to default to 1 second during extended periods of inactivity.

C.4.4.4.1.3. Calculation of the scheduler random parameter. The parameter schedint depends on queue lengths and average concatenated frame transmit duration as follows:

$$\text{schedint} = \text{Fsched} * \text{Ttrans}, \; \text{min} \le \text{schedint} \le \text{max}$$

where:

Ttrans = Average concatenated frame transmit duration.

Fsched = Scheduling Factor.

min = 3 seconds; max = 20 seconds.

schedint shall be recomputed after every transmission by the DTE.

C.4.4.4.1.4. Calculation of the scheduling factor. The scheduling factor, Fsched, is based on three other factors: 1) the Partition Factor, Fpart, 2) the Topology Factor, Ftop, and 3) the Load Factor, Fload as follows:

$$\text{Fsched} = (\,(\text{T}*\text{Ftop})\,(\text{Fload})\,)/(\,(\text{P}*\text{Fpart}) + \text{D}\,)$$

such that min<=Fsched<=max

where:

	FH	SC
T =	1	2
P =	3	3
D =	7	7
min =	1	1
max =	20	20

and

FH = Frequency Hopping Mode
SC = Single Channel Mode
T = Topology Coefficient
P = Partition Coefficient
D = Damping Coefficient

Fload and Fsched are recomputed after every transmission attempt by the local station. Fpart and Ftop are computed after a topology update is generated or received. The T, P and D coefficients represent a compromise between high throughput and low delay, while promoting channel stability. The increase in coefficient T for single channel mode is compensating for the hidden terminal effect in multi-hop fixed frequency networks. The minimum value of 1 second has a throttling effect on the amount of time a station must wait between channel access attempts in a heavily congested, large network with multiple relayers.

C.4.4.4.1.4.1. Calculation of the Partition Factor. The Partition Factor is computed in a fully distributed manner by each node in the net. Partition Factor takes into account the one-hop connectivities experienced by each node in the network and is a measure of the connectivity between a node's neighbors. When a node's neighbors are strongly interconnected, i.e., the neighbors can hear each other, traffic will be routed directly between neighbors without a need for the node in question to relay the traffic. However, when a node's neighbors are weakly interconnected, for example the neighbors cannot hear each other, the node in question will see an increase in traffic due to relaying between neighbors.

The Partition Factor takes values between 1 for a strongly connected network and 7 for a weakly connected network. In the case of a higher partition factor the channel access scheduling interval is decreased, (the scheduling rate is increased), at the node doing the calculation, to meet the need for a higher percentage of the channel capacity to handle increased relay traffic. The Partition Factor for a non-relay node is set to 1 without executing the following algorithm.

Partition Factor is computed after a topology change is detected.

ALGORITHM: Calculation of the Partition Factor, Fpart, at node *j*.

1. Set the number of neighbors to zero, set the number of broken links to zero.

2. For each node *i* in the network, if there is a one hop link from *i* to *j* then:

2a. Increment the number of neighbors.

2b. For each node *k* in the network, if there is a one hop link from *k* to *j* and there is no one hop link from *i* to *k* then increment the number of broken links.

3. max links = # of neighbors * (# of neighbors -1)

4. If max links less than 1 then max links = 1.

5. Fpart = broken links * 6 / max links.

6. Fpart = Fpart + 1.

7. FINISH.

The constant 6 in algorithm step 5 establishes a set of 7 different levels of expected relaying. Nodes that are expected to do a significant amount of relaying (because their neighbors are not strongly connected) will receive the value 7 for Fpart, which shortens their scheduling interval per the algorithm in C.4.4.4.1.4. Nodes that are not expected to do a significant amount of relaying (because their neighbors are fully connected, thus reducing the amount of relaying required and providing a number of nodes to share relaying responsibilities) will receive the value 1. This lengthens their scheduling interval Nodes whose neighbors are neither strongly nor weakly connected are assigned a value between 1 and 7, exclusive, that depends on the degree of connectivity between the node's neighbors. This provides a total range of 7 values to bias a node's scheduler, depending upon the degree of relaying that a node is expected to do.

C.4.4.4.1.4.2. Calculation of the Topology Factor. The Topology Factor is computed in a fully distributed manner by each node in the net. This algorithm computes the ratio of the number of nodes that are one and two hops away from the node doing the computation, to the number of neighbors of the node doing the computation. For the Topology Factor Calculation the following applies: (1) a neighbor is defined as a node which is one hop away from the node doing the calculation, and (2) a neighbor node can also be categorized as a two-hop away node, from the node doing the computation, as long as there is a way of reaching that node in two hops (i.e., a neighbor node can simultaneously be listed, if it meets the criteria, as a two-hop away node). Since this is computed at each node in a fully distributed manner, this enables each node to evaluate its own situation in the network with respect to competition for the shared channel. A node with a high ratio of nodes within two hops to number of neighbors should use a longer scheduling interval due to the fact that neighbor nodes will have to handle the relaying traffic between the node under consideration and all other nodes in the net. Also, neighbor nodes will experience a higher ratio of receive collisions since the node in question and the nodes two hops away are "hidden" from each other and thus cannot cooperate well in the channel access process.

Topology Factor takes values from 3 for a low ratio of nodes within two hops to neighbors, and 40 for a high ratio of nodes two hops to neighbors. In cases of higher Topology Factor values it is desirable to use a longer channel access scheduling interval to reduce the occurrence of channel access contention and collisions.

Topology Factor is computed after a topology change is detected.

ALGORITHM: Calculation of Topology Factor, Ftop, for node *j*

1. Set "number of neighbors" to zero.

2. Set "hearing within two hops" to zero.

3. For each node *i* in the net, if there is a one hop link from unit *i* to unit *j* then

3a. Increment the number of neighbors.

3b. Hearing within two hops = hearing within two hops + (number of neighbors of unit *i*) - 1. (don't count unit *j*).

4. If the number of neighbors is zero then set Ftop = 10. Go to step 10.

5. Hearing within 2 hops = hearing within 2 hops + # of neighbors.

6. Hearing within 2 hops = (hearing within 2 hops * 6)/4.

7. Ftop = hearing within two hops/number of neighbors.

8. If Ftop is less than 3 then Ftop = 3.

9. If Ftop is greater than 40 then Ftop = 40.

10. FINISH.

The constants 6 and 4 in algorithm step 6, and the constant 10 in algorithm step 4 are the default values which, when used in the Fsched equation (see C.4.4.4.1.4), restrict the channel access opportunities to a stable region offering good throughput and delay characteristics. The constants 6 and 4 in algorithm step 6 throttle the scheduler as the number of nodes increases, which preserves throughput. The values 6 and 4 give more range when using integer division than the values 3 and 2, respectively.

C.4.4.4.1.4.3. Calculation of the Load Factor. The Load Factor is computed in a fully distributed manner by every node in the net. In the transmission header, one byte is dedicated to transmitting the number of messages at the highest priority level remaining in the information frame queue. The four most significant bits (MSB) indicate the level of the highest priority message. The three least significant bits (LSB) indicate the number of frame concatenation's required to transmit all of the messages at the above priority level. The four LSB are quantized as shown in Table C-1.

TABLE C-1. Calculation of the load factor.

Number of Concatenated Frames Required	Bit Pattern (LSB is on the right)
0.0	0 0 0 0
0.0 (exclusive) - 0.5 (inclusive)	0 0 0 1
0.5 (exclusive) - 1.0 (inclusive)	0 0 1 0
1.0 (exclusive) - 2.0 (inclusive)	0 0 1 1
2.0 (exclusive) - 3.0 (inclusive)	0 1 0 0
3.0 (exclusive) - 4.0 (inclusive)	0 1 0 1
4.0 (exclusive) - 5.0 (inclusive)	0 1 1 0
> 5.0	0 1 1 1

The Load Factor takes on values such that 1.0 < Fload < 18.0. The minimum value of 1.0 places an upper limit on the speed of the scheduler per the Fsched equation in C.4.4.4.1.4. The value of 18.0 provides a useful range for adaptation of the scheduler due to differing traffic loads, and is divisible by 2 and 3, resulting in integer ranges for the three different precedence values. Higher values of the Load Factor indicate that the node has shorter queues of equal or lesser priority. In cases of high load factor, it is desirable to increase the scheduling interval to give neighboring nodes with higher priority or longer queues of equal priority more opportunities to transmit. The Load Factor is calculated after every expiration of the scheduler, prior to calculation of the next expiration.

ALGORITHM: Calculation of the Load Factor, Fload.

1. Determine the number of unique neighboring node priority levels broadcast by all the nodes including this one. This data is taken from the last transmission received from each neighboring node.

2. Divide the interval 0.0 to 18.0 into equal segments, one per unique announced priority level. The first segment (0.0 to 9.0 for two levels) is allocated to the highest priority traffic. Define the Segment Offset as the lower bound of the chosen segment. For two precedence levels, the Segment Offset is 0.0 for the highest precedence and 9.0 for the Lowest Precedence. Define the Segment width equal to 18.0/Number of precedence levels. For all three precedence levels, each precedence level has a segment width of 6.0

3. Each segment is subdivided into n unique levels where n is the number of unique quantized concatenated frame lengths reported by the neighboring nodes and the node doing the computation. In the case of only one length, all nodes use the midpoint of the segment. In the case of multiple lengths, these lengths are ordered from longest to shortest (1 -> n). In the following computation of Load Factor, a node would use a value of m determined by its position in that ordering. All nodes with the longest quantized length use the value 1, while those with the shortest use the value n for variable m in the following equation:

   Load Factor = Segment offset + (Segment width*m)/(n+1)

   where

   Segment Offset is the Lower bound of the segment chosen by precedence level from step 2.

   Segment Width is the maximum Load Factor (18) divided by the number of unique precedence levels

   n is the number of unique quantized queue lengths.

   m is this nodes positioning within the ordering of quantized queue lengths.

TABLE C-2. Calculation of the load factor -- example 1.

Node Number	Highest Precedence	Quantized Queue Length	Load Factor
1	Routine	0 0 0 1	12.0
2	Routine	0 0 0 1	12.0
3	Routine	0 0 1 1	6.0
4	Routine	0 0 1 1	6.0

All nodes compute the load factor in the following manner.

1. There is only 1 unique priority level (Routine).

2. The Segment is determined to encompass the whole range 0->18.

3. The Segment Offset is the lower bound (0).

4. The Segment Width is the entire range (18).

5. Two unique Quantized Queue Lengths are noted. The value of n is set to 2.

6. The unique Quantized Queue Lengths are ordered from longest to shortest (3,1).

7a. Nodes 1 and 2 note that their positioning in this sequence is 2 and set m to 2.

7b. Nodes 3 and 4 note that their positioning in this sequence is first and set m to 1.

8a. Nodes 1 and 2 compute their load factor from the equation.

Load_Factor = Segment Offset + (Segment Width*m)/(n+1)

= 0 + (18 * 2) / (2+1) = 12

8b. Likewise, Nodes 3 and 4 do the Load Factor computation.

Load_Factor = 0 + (18 * 1) / (2+1) = 6

TABLE C-3. Calculation of the load factor -- example 2.

Node Number	Highest Precedence	Quantized Queue Length	Load Factor
1	Routine	0 0 0 1	13.5
2	Routine	0 0 0 1	13.5
3	Urgent	0 0 1 0	6.0
4	Urgent	0 0 1 1	3.0

All nodes compute the load factor in the following manner.

1. There are two unique precedence levels (Urgent and Routine).

2. The load Factor Range is divided into two segments 0-9, 9-18. The segment 0-9 is reserved for Urgent, while the segment 9-18 is reserved for Routine.

3. The Segment Offset is the lower bound of the segment. The Segment Offset is 0 for Urgent and 9 for Routine.

4. The Segment Width for both precedence levels is the entire range (0->18) divided by the number of precedence levels. Segment Width = 18/2 = 9.

Nodes 1, 2 perform the following computations:

5. There is only one Quantized Queue Length. Thus, n is equal to 1 and since there is only 1 length both nodes use the first position in the sequence and set m to 1.

6. Load Factor = Segment Offset + (Segment Width*m)/(N+1)
   = 9 + (9 * 1) / (1+1) = 13.5

Nodes 3,4 perform the following computations.

7. The unique Quantized Queue Lengths are ordered from longest to shortest (3,2). There are two unique lengths which sets the value of n to 2.

8. Node 3 has a length of 2 which occupies position 2 in the ordering of step 7. Because is occupies position 2, the value of m is set to 2.

   Load_Factor = Segment Offset + (Segment Width*m)/(n+1)
   = 0 + (9 * 2) / (2+1) = 6

   Node 4 has length of 3 which occupies position 1 in the ordering of step 7. Node 4 sets it's value of m to 1.

   Load_Factor = Segment Offset + (Segment Width*m)/(N+1)
   = 0 + (9 * 1) / (2+1) = 3

C.4.4.4.1.5 Immediate mode scheduling. The average scheduling interval of the continuous scheduler is a factor in determining intranetwork end-to-end delay. In a lightly loaded network the average end-to-end delay will not be less than the average scheduling interval. In large nets this contributes to unnecessarily large end-to-end delay under conditions of low input load.

This situation is corrected by use of "immediate mode" scheduling under certain specific conditions.

The problem mentioned above is most obvious in large nets under conditions of light load. The Topology Factor incorporates network size in computing the scheduling interval increases as network size increases. However, in large nets under conditions of light input load channel utilization is low, yet end-to-end will be unnecessarily large due to the average scheduling interval of the continuous scheduler.

This situation is corrected using "immediate mode" scheduling as follows:

a. If the message is Type 1 and requires a coupled acknowledgment, set Tc to 0.0 seconds and initiate an immediate channel access attempt. If the DCE is busy, implement the 1-persistent DTE-DC channel access protocol and transmit when the busy period ends. All stations receiving this transmission will suspend their Tc timers and observe the Type 1 timing for coupled acknowledgments.

b. If the scheduler expires and there are no concatenated frames awaiting transmission, set Immediate Mode true. Compute and start the next random interval of the continuous scheduler (Tc).

c. When an information PDU arrives for transmission and Immediate Mode is true, compute a scheduling interval as follows:

$$Tc = 100 \text{ msec}$$

d. When this is done, Immediate Mode is reset to false. The previously scheduled Tc is to be canceled. The 100 msec allows an opportunity for messages which have been segmented/fragmented/received to be piggy-backed into the same series of concatenated frames. This increases efficiency without imposing delay.

e. When the scheduler expires due either to the Tc scheduled as a result of the immediate mode operation or due to normal continuous operation, and I-frame(s), S-frame(s), U-frame(s) or a frame concatenation are pending, perform concatenated frame processing as normally is done. Compute and start the next random interval of the continuous scheduler (Tc) in the normal manner.

f. The Tc interval timer set as a result of immediate mode operation is to be suspended and resumed for voice operation as is done for continuous mode operation.

C.4.4.4.2 RE-NAD network access. When the precedence level of the transmission changes, the DTE shall set the precedence level of the new transmission. This precedence level will correspond to the frame with the highest precedence value within the series of concatenated frames.

C.4.4.4.3 Network busy sensing and receive status. The presence of multiple stations on a single random access communications network requires voice/data Network Busy Sensing and the use of network access control to reduce the possibility of data collisions on the net. The combined Data and Voice Nets require cooperation between the DTE (DMTD or C.4I system) and the DCE.

The DCE indicates the presence of receive data and voice by signaling the following conditions:

a. Data being received,

b. Voice operation,

c. Idle/Transmission completed,

d. Data being transmitted.

The transmission of data by the DTE is allowed only in the Idle/Transmission completed state.

C.4.4.5 Deterministic Adaptable Priority-Network Access Delay (DAP-NAD). DAP NAD is a method of generating Network Access Delays to control network accesses which provides every subscriber with an equal opportunity (when considering multiple access periods and equal message priorities) to use a radio/wireline net. It is deterministic in that every subscriber has an opportunity to access the network and given the device, network, and protocol parameter settings, the maximum time for network access can be calculated.

The mechanism for providing equal network access is to give the first "access opportunity" (the time at which a subscriber may transmit a message if one is available) to a different subscriber at each "network access period" (the time between message transmissions when all subscribers are determining when to transmit) and to give later access opportunities to all other subscribers in sequence. Each subscriber is assigned a Subscriber Rank that is in the range of 1 to the Number of Subscribers (NS in the equations that follow). During the first network access period, subscriber number 1 is given the first access opportunity, subscriber number 2 is given the second access opportunity, subscriber number 3 the third access opportunity, etc. After the last subscriber has been given an access opportunity, subscriber number 1 is again given an access opportunity, followed by subscriber number 2, etc. This continues until a subscriber transmits a message. The subscriber that transmits the message shall increment the First Subscriber Number subfield contained in the last message it received and place the number in the First Subscriber Number subfield of the Transmission Header. The very next access period (the first DAP-NAD time slot following the message transmission) is reserved, such that any node can interrupt the network in case the network priority is lower than the precedence of the message they have to transmit. This reserved slot is only used when the network is in Priority or Routine mode. All nodes having messages to transmit with a precedence that is greater than the current network priority would transmit a short Urgent control frame in the reserved slot. Upon receipt of this Urgent control frame or detection of a network busy condition during the reserved slot, all receiving nodes would assume that the network priority had gone to Urgent and act accordingly.

In this manner, transmissions in the reserved slot would serve to interrupt the operation of a network operating at Priority or Routine causing it to elevate to Urgent mode. The next station authorized to access the network is the First Station Number specified in the Transmission Header of the transmission that occurred before reverting to Urgent mode. Each subscriber calculates different NAD times for each network access period. There are three network priority modes; urgent, priority and routine. The reserved slot is not provided when the network is in the Urgent mode. The calculation of the NAD times are discussed in the following paragraphs.

a. Network in Urgent Mode. The first NS number of access opportunities of a network access period are reserved for subscribers that have an urgent message awaiting transmission. Those subscribers that do not have any urgent messages awaiting transmission must wait for at least the NS+1 access opportunity before they can transmit. The next NS number of access opportunities of the network access period are reserved for subscribers that have a priority (or an urgent if one has become available since the previous access opportunity) message awaiting transmission. Those subscribers that have only routine messages awaiting transmission must wait for at least the 2NS+1 access opportunity before transmitting. Those subscribers that have any messages awaiting transmission, regardless of priority, by the 2NS+1 access opportunity can transmit when their calculated access opportunity arrives.

b. Network in Priority Mode. The first NS+1 access opportunities are reserved. Access opportunities 2 through NS+1 are reserved for subscribers that have an urgent or priority message awaiting transmission. Those subscribers that only have routine messages awaiting transmission must wait for at least NS+2 access opportunity before they can transmit. Those subscribers that have any messages, regardless of priority, awaiting transmission by the NS+2 access opportunity can transmit when their calculated access opportunity arrives. The very first network access period following completion of the transmission while in Priority mode shall be reserved for any station with an Urgent message to notify all other subscribers to revert back to Urgent network mode. After reverting to Urgent mode, the subscriber with the station number matching the First Subscriber Number in the Transmission Header of the transmission completed just before the reserved slot shall have the first network access opportunity. The network shall then remain in the Urgent mode until all stations have had an opportunity to access the network.

c. Network in Routine Mode. Only the first access opportunity is reserved. After that, any subscriber that has a message, regardless of priority, can transmit when their calculated access opportunity arrives. The very first network access period following completion of the transmission shall be reserved for any station with an Urgent or Priority message to cause the network to go to Urgent mode. If no station transmits during the reserved slot, the network remains in the mode designated by the Data Link Precedence field in the Transmission Header provided by the last station accessing the network. Routine mode remains in effect until a message is transmitted.

C.4.4.5.1. DAP-NAD Information Field. To allow for rapid recovery (resynchronization) to the DAP-NAD mechanism when messages are not received correctly due to noise, etc., and to provide subscribers information about the priority of a message, a DAP-NAD Information Field has been added to the Transmission Header. This field defines the next access opportunity. This field is present in all physical frames. This field contains the First Subscriber Number subfield which contains the number of the subscriber that is to have the first network access opportunity at the next network access period (the one immediately following this transmission). The number of the subscriber that has the first network access opportunity is the variable FSN in the equations below. The DAP-NAD Information Field also contains the Data Link Precedence subfield which indicates the highest priority of any message that is contained in the physical frame. It shall contain the value 0 if an urgent message is in the frame, 1 if a priority but no urgent message is in the frame and 2 if neither an urgent or priority message is in the frame. The Type 1 acknowledgment sent in response to a transmission will use the same Data Link Precedence and First Subscriber Number as used in the original message to which the acknowledgment applies. The variable NP in the equations below shall be set equal to the content of this subfield for the next network access period. If the transmission contained multiple frames, the variable NP is set equal to the highest value in any of the frames. If network busy is detected in the reserved network access period, the network reverts to the Urgent mode regardless of the setting in the Data Link Precedence subfield.

C.4.4.5.2 DAP-NAD Equations. A sequence of NADs for each network access period is generated. A subscriber may transmit a message(s) when the time following the Timeout Period equals any one of the terms (NAD values) in the sequence. Equation 1 is used by each subscriber to calculate its DAP-NADs.

Equation 1: $NAD_n = F_n * Net\_Busy\_Detect\_Time + Max(0, F_n-1) * DTETURN$
for n=1 to 4

NAD_n is the *n*th term in the sequence of NADs that are associated with a subscriber during a network access period. Each term (NAD_1, NAD_2, NAD_3, etc.) is a point in time (a delay following the Timeout Period) at which a subscriber may begin transmitting. If a subscriber does not begin transmitting at one term (e.g. NAD_2), it must wait until at least the next term (e.g. NAD_3) before it can begin transmitting. For the DAP-NAD method, the values of the terms calculated by a subscriber will be different than the values of the terms that are calculated by all of the other subscribers (no two subscribers will have terms with the same values). Also, the values of the terms calculated by a subscriber for one network access period will be different than the values of the terms calculated by that subscriber for the next network access period. F_n is *n*th term in a sequence of factors that, when used in conjunction with DTETURN and Net_Busy_Detect_Time, yields the *n*th NAD term. F_n is an integer calculated per equation 2.

Equation 2; F_n $= F_1 + (n\text{-}1)NS$ for n=1 to 4

F_n is the nth term in a sequence of factors. F_1 is the first term in the sequence and is the base from which all the other terms are calculated. It is calculated per equation 3. NS is the number of subscribers on the network and is the common difference between the terms of the sequence. The variable n is an integer and has a range of 1 to infinity.

Equation 3: F_1 $= F_{min} + I + P * NS$

F_1 is the first term in the sequence of factors. The first term that a subscriber can have is the minimum factor (F_{min}) plus the interrupt factor (I) plus an offset determined by priority of messages awaiting transmission. F_{min} is calculated using equation 4. P is the factor that accounts for message priority. It is calculated using equation 5. Interrupt factor I is computed using equation 6.

Equation 4: F_{min} $= SN - FSN$ if $SN \geq FSN$, else $F_{min} = NS+SN-FSN$

F_{min} is the minimum possible factor that a subscriber could have if message priority and network priority mode were ignored. SN is the number of the subscriber. It is an integer, has a range of 1 to NS, and is assigned at communications initialization. FSN is the number of the subscriber that has the first network access opportunity for the present network access period. It is set equal to the value received in the DAP-NAD information field of the Transmission Header of the last transmission on the net.

Equation 5; P $= MP - NP$ if $MP \geq NP$. else $P = 0$

P is the factor that accounts for priority of messages awaiting transmission. It is used to generate the offset to add to F_{min} to generate F_1. MP is a variable indicating the highest priority of any messages awaiting transmission. It shall have a value of 0 if there are any urgent messages awaiting transmission, the value 1 if there are any priority messages and no urgent messages awaiting transmission, and the value 2 if there are no urgent or priority messages awaiting transmission. NP is a variable indicating the highest priority of any messages contained in the last transmission on the network. It shall have the value 0 if an urgent message was in the last transmission, 1 if a priority but no urgent message was in the last transmission, and 2 if neither an urgent or priority message was in the last transmission.

Equation 6: I $= 0$ if NP is 0, else $I = 1$

I is a factor that provides a slot for stations to interrupt the network when the network is not already in Urgent mode.

C.4.4.5.3 Initial Condition State. The above DAP-NAD operations and equations only apply to subscribers after they are on-line and have received a message. A subscriber that has just come on line and has not yet received a message is not in synchronism with other subscribers (this subscriber has not yet started any timers and if it had, they would not have been started at the same time as other subscribers = timers). These subscribers shall be considered to be in the initial condition state. Regardless of what causes a subscriber to be in the initial condition state, transmissions must be delayed by at least the time specified by equation 7 while in that state.

Equation 7: INAD = TP + ((3 * NS) + 1) * Net_Busy_Detect_Time + (3 * NS) * DTETURN

INAD (Initial condition state Network Access Delay) is the minimum time that a subscriber must delay transmission of a message after it has become capable of receiving and transmitting messages, but no more than 20 seconds. The TP in the equation shall be a worst case TP, i.e., as if there had just been a Type 1 message on the network that required acknowledgment and was addressed to 16 subscribers on the net.

C.4.5. Voice/data network sharing. A station may support this protocol on a network where both voice and data transmissions are allowed to occur. When operating in a mixed voice and data network, voice and data network sharing shall operate in the following manner:

a. A receive operation shall be considered a voice reception unless a valid synchronization pattern is identified. A receive operation that is less than 0.75 seconds in length shall be considered a noise burst instead of a voice reception. See Section 6, Notes, (6.3.2.2.2) for additional information.

b. The network shall be synchronized based on RHD and TP timers, which are driven only by data transmissions and receptions. Voice receptions and noise bursts shall not be used for resynchronizing network timers.

c. A station shall not transmit during a noise burst or a voice reception. After completion of a voice reception, a station shall wait at least TURN milliseconds before initiating transmission. When voice/noise reception begins and ends during a Type 1 acknowledgment sequence, an acknowledging station will begin transmission only at the beginning of its individual RHD and will not begin transmission after the start of its RHD period.

d. After completion of a voice reception, operation of the P-NAD network access scheme shall be reinitiated if P-NAD is being used. P-NAD consists of a sequence of NAD slot groups. Within each NAD slot group there is one NAD slot assigned to each station and one slot assigned to the station that transmitted last. After a voice reception is completed, the current, partially-completed NAD slot group and the next complete NAD slot group shall be used only by stations with urgent-precedence data transmissions. The NAD slot group after these groups shall be used only by stations with urgent-precedence or priority-precedence data transmissions. Subsequent NAD slot groups may be used by any station. This preserves the intent of P-NAD, which is to deterministically avoid collisions and to ensure that high-precedence traffic is always transmitted first.

e. RHD and TP timers shall not be suspended or resumed as a result of voice receptions.

f. Data link protocol timers shall be suspended and resumed as a result of voice receptions.

g. The Intranet layer timers shall not be suspended and resumed as a result of voice receptions.

h. Relative priorities of voice and data on the network shall be adjusted by selectively enabling or disabling physical and/or data link concatenation for a station. Concatenation may be disabled to give priority to voice and may be enabled to give priority to data.

APPENDIX D

COMMUNICATIONS SECURITY STANDARDS

D.1. General.

D.1.1 Scope. This appendix describes the COMSEC interoperability parameters for DMTD and interfacing C^4I systems. It defines the technical requirements for backward-compatible (traditional) and forward-compatible (embedded) interface modes. See classified Appendix D-2 for additional information.

D.1.2 Application. This appendix is a mandatory part of MIL-STD-188-220. The information contained herein is intended for compliance.

D.1.3 Interoperability. This appendix cannot guarantee the user end-to-end interoperability. The selection of COMSEC and signaling is a function of communications media. Traditional COMSEC equipment is specific to communications media and may not be compatible due to signaling differences. The systems integrators and systems planners must ensure that compatible media and signaling are chosen if interoperability is desired. This COMSEC specification will provide for interoperability of the underlying encryption algorithm.

D.2. Applicable Documents.

a. (U) ON431125 WINDSTER Cryptographic Standards

b. (U) DS-68 INDICTOR Cryptographic Standards

D.3. Definitions. Refer to Appendix A.

D.4. General Requirements. The backward-compatible mode applies when link encryption is provided by external COMSEC devices. These external COMSEC devices may be standalone equipment (such as the VINSON and KG-84) or communications equipment with built-in COMSEC. The forward-compatible mode shall apply for all DTE subsystems with embedded COMSEC. The backward-compatible mode may also be emulated using embedded COMSEC devices.

D.5. Detailed Requirements.

D.5.1 Traditional COMSEC transmission frame. The traditional COMSEC transmission frame shall be composed of the following components, as shown in Figure D-1. Figure D-1 provides additional detail to Figure 4a.

a. COMSEC Bit Synchronization

b. COMSEC Frame Synchronization

c. Message Indicator

d. Phasing

e. Transmission Synchronization (see 5.2.1.3).

f. Data Field (including Transmission Header)

g. COMSEC Postamble

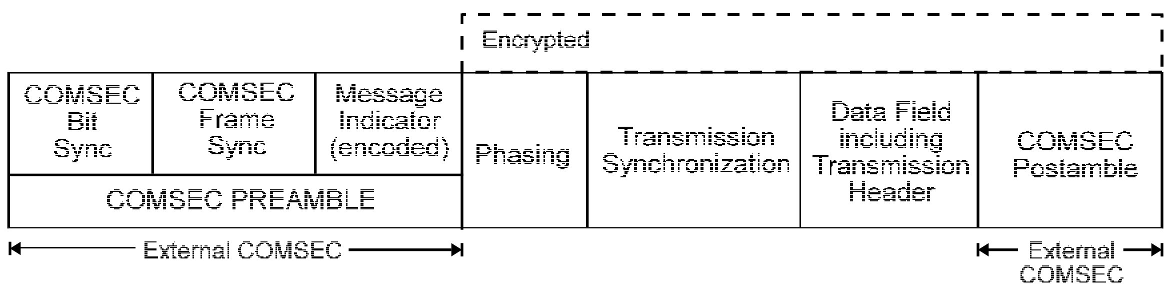


FIGURE D-1. Traditional COMSEC transmission frame structure.

D.5.1.1 COMSEC preamble field. The COMSEC preamble field shall consist of three components: a COMSEC bit synchronization subfield, a COMSEC frame synchronization sub-field, and a Message Indicator (MI) subfield. This field is used to achieve cryptographic synchronization over the link.

D.5.1.1.1 COMSEC bit synchronization subfield. This subfield shall be used to provide a signal for achieving bit synchronization and for indicating activity on a data link to the receiver. The duration of the COMSEC bit synchronization subfield shall be selectable from 65 milliseconds to 1.5 seconds. The COMSEC bit synchronization subfield shall consist of the data-rate clock signal for the duration of the subfield.

D.5.1.1.2 COMSEC frame synchronization subfield. This subfield shall be used to provide a framing signal indicating the start of the encoded MI to the receiving station. This subfield shall be 465 bits long, consisting of 31 Phi-encoded bits, as shown in Figure D-2. The Phi patterns are a method of redundantly encoding data bits. A logical 1 data bit shall be encoded as Phi(1)= 111101011001000, and logical 0 data bit shall be encoded as Phi(0)= 000010100110111. A simple majority voting process may be performed at the receiver to decode the Phi-encoded frame pattern to its original format.

LSB	MSB
1111111111111111111111111111110	

FIGURE D-2. COMSEC frame synchronization pattern for Phi encoding.

D.5.1.1.3 Message Indicator subfield. This subfield shall contain the COMSEC-provided MI, a stream of random bits that are redundantly encoded using Phi patterns. Cryptographic synchronization is achieved when the receiver acquires the correct MI.

D.5.1.2 Phasing. This field shall be a string of alternating ones and zeros, beginning with a one, sent by the DTE. The length of this field is between 0 and 10,000 milliseconds. Phasing is further described in C3.2b.

D.5.1.3 Transmission synchronization field. This field, consisting of the frame synchronization subfield, optional robust frame format subfield, and the TWC subfield, shall be as defined in 5.2.1.3.1.4.

D.5.1.4 Data field. This field, including Transmission Header as defined in 5.3.1, shall be as defined in 5.2.1.4.

D.5.1.5 COMSEC postamble field. This field shall be used to provide an end-of-transmission flag to the COMSEC at the receiving station. This will be automatically performed by the COMSEC key generator. Refer to 0N431125, WINDSTER Cryptographic Standards, or DS-68, INDICTOR Cryptographic Standards, as appropriate.

D.5.1.6 COMSEC algorithm. The COMSEC algorithm shall be backward-compatible with VINSON equipment. Refer to 0N431125, WINDSTER Cryptographic Standards.

D.5.1.7 COMSEC modes of operation. The COMSEC shall be operated in Mode A. The rekey functions shall be performed through the use of KY-57 rekeys for backward compatibility. Refer to 0N431125, WINDSTER Cryptographic Standards.

D.5.2 Embedded COMSEC transmission frame. The embedded COMSEC transmission frame shall be composed of the following components, as shown in Figure D-3:

a. Phasing

b. Frame synchronization

c. Optional Robust Frame Format

d. Message Indicator

e. Transmission word count

f. Data Field

g. COMSEC Postamble

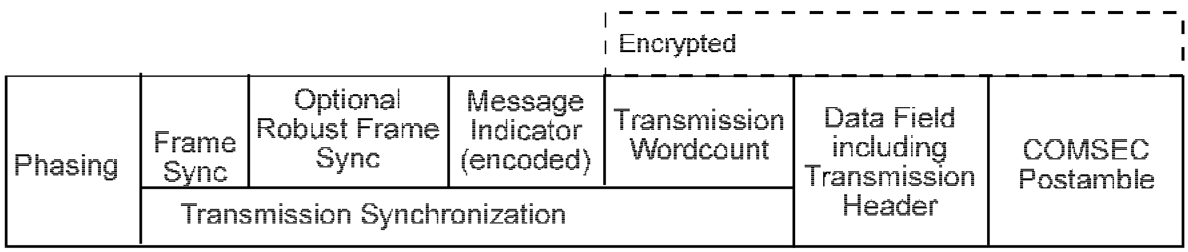


FIGURE D-3. Embedded COMSEC transmission frame structure.

D.5.2.1 Phasing. This field shall be a string of alternating ones and zeros, beginning with a one, sent by the DTE. The length of this field is between 0 and 10,000 milliseconds. Phasing is further described in C3.2b.

D.5.2.2 Frame synchronization subfield. This subfield shall be either the Robust Frame Synchronization subfield defined in 5.2.1.3.1.2 or the Frame Synchronization subfield defined in 5.2.1.3.1.1. In either case frame synchronization is to be provided for both the message frame and the COMSEC.

D.5.2.3 Robust Frame Format subfield. When the Robust Frame Synchronization subfield is used, the Robust Frame Format subfield defined in 5.2.1.3.1.2 also shall be used. The Robust Frame Format subfield shall not be used when the Robust Frame Synchronization subfield is not used.

D.5.2.4 Message Indicator field. This field shall contain the MI, a stream of random data that shall be encoded using Golay, as defined in 5.3.14.1 and 5.3.14.2. Cryptographic synchronization is achieved when the receiver acquires the correct MI. The COMSEC shall provide the MI bits. For backward compatibility, these MI bits must be redundantly encoded using Phi patterns, as described in D.5.1.1.2.

D.5.2.5 Transmission word-count subfield. This subfield shall be as defined in 5.2.1.3.1.4.

D.5.2.6 Data field. This field, including Transmission Header as defined in 5.3.1, shall be as defined in 5.2.1.4.

D.5.2.7 COMSEC postamble field. This field shall be used to provide an end-of-transmission flag to the COMSEC at the receiving station. The flag shall be a cryptographic function and may be used by the data terminal as an end-of-message flag as well.

D.5.2.8 COMSEC algorithm. Refer to 0N431125, WINDSTER Cryptographic Standards.

D.5.2.9 COMSEC modes of operation. COMSEC shall be operated in Mode A for all applications. The rekey functions will be performed through the use of KY-57 rekeys for backward-compatibility and will be performed through over-the-air-rekeying (OTAR) techniques for forward compatibility. Rekey signaling for OTAR must be supplied by the host equipment. Refer to 0N431125, WINDSTER Cryptographic Standards.

APPENDIX E

CNR MANAGEMENT PROCESS

E.1. General.

E.1.1 Scope. This appendix describes the management processes associated with the data link and network layer. Since the tactical network using CNR may not be fully connected and since it is critical that all stations are provided compatible operating parameters, an Exchange Network Parameters (XNP) message has been defined. XNP messages that are transmitted within Type 1 UI frames, can be relayed, allow disconnected stations to participate fully in the network, and can be used to change network parameters dynamically.

E.1.2 Application. This appendix is not a mandatory part of MIL-STD-188-220.

E.2. Applicable Documents. None

E.3. Network Configuration. The CNR management process defined herein covers both centralized and distributed operations. The procedure for negotiation of parameters varies with network configuration. In a centralized network, a single network controller manages the network. In a distributed network, the network is managed by multiple network controllers. It is possible for any number of stations, even all stations, in an established network to be network controllers.

It is desirable that all stations be capable of performing the functions of network controller. The designation of network control station(s) will be done by a network authority. A configuration parameter or an operator command either at initialization or during normal operation times, may inform the station of its network control responsibility.

E.3.1 Centralized Network Control. Centralized Network Control requires that one network controller manage and control all aspects of the network. Although all stations within a network are expected to be capable of performing the functions of the network controller, only one station is designated the network controller at any one time. Access parameters may only be obtained from the one designated network controller.

E.3.2 Distributed Network Control. Distributed Network Control allows multiple stations to share the functions of network controller. This is especially useful in disconnected networks where unique access parameters, such as time slots for deterministic network access, are not required. Access parameters may be obtained from any station acting as the network controller. Control of parameters is maintained by interactions between the participating network controllers.

E.4. Exchange Network Parameters (XNP) Message. XNP messages have been designed to provide CNR management capabilities. However, they are not required if the stations on the network have been configured with data link addresses and operating parameters.

E.4. Exchange Network Parameters (XNP) Message. XNP messages have been designed to provide CNR management capabilities. However, they are not required if the stations on the network have been configured with data link addresses and operating parameters.

E.4.1 XNP Message Structure. XNP messages are composed of a one-octet Version Number field (set to 0), an optional Forwarding Header followed by the actual XNP message and one or more data blocks as shown in Figure E-1.

Most Significant Octet<————————————————————>Least Significant Octet

Version Number	Forwarding Header (Optional)	XNP Message (Identification plus basic information fields)	One or More XNP Data Blocks

FIGURE E-1. XNP message format.

Detailed formatting of the Forwarding Header, each XNP Message and each Data Block is described in the following paragraphs and tables. The Forwarding Header, each XNP Message and each Data Block consists of data fields. The data fields may be one or more octets in length and may be value coded or bit mapped. When the data field size exceeds one octet, octets are transmitted from the most significant octet (low number) to the least significant octet (high number). Bit mapping uses each bit individually in an on/off representation such that multiple values may be represented by each octet. Bit 0 always represents the least significant bit (2^0). Figure E-2 depicts an example 4-octet data field.

	Most Significant Octet<————————>Least Significant Octet							
Octet	**1**		**2**		**3**		**4**	
Bit	7	0	7	0	7	0	7	0
Bit	31	24	23	16	15	8	7	0
	Most Significant Bit<————————————>Least Significant Bit							

FIGURE E-2. Example 4-octet XNP data field.

Undefined bits shall be set to zero on transmission and ignored on receipt. Undefined values are invalid. The processing of XNP messages containing undefined/invalid values shall be:

a. Ignore any undefined bits in a bit map.

b. If the Version Number is invalid or unsupported, discard the XNP message.

c. If any field in the Forwarding Header is invalid, discard the XNP message.

d. If the Message Number field is invalid, discard the XNP message.

e. If the Block Number field is invalid in any XNP message, discard the block and continue processing the XNP message.

f. If the Length field is invalid in any Data Block (i.e., the value indicates that there are more octets than actually exist in the XNP message), discard the rest of the XNP message but act on the preceding blocks if possible.

h. If any other field is invalid in any Data Block, discard the data block and continue processing the XNP message.

g. If any other field is invalid in an XNP message, discard the XNP message.

E.4.1.1 Forwarding Header. A station joining a network might not have knowledge of the topology and might be unable to contact all stations in the network being joined (e.g. stations might be behind obstacles or out of range). The Forwarding Header provides a means for a joining station to make use of an adjacent station which has access to the entire network via Intranet Relay.

Relay assistance is required for the joining process in a centralized or distributed network when the joiner is unable to directly communicate with the network controller, and also used in distributed networks to ensure adequate distribution of the Hello message. The joining station fills in this header to request relay assistance by an established station. The selected established station distributes these messages using appropriate Intranet Relaying techniques. Any and all responses go back to the selected established station who then forwards to the joiner. When an established member of a network receives an XNP message that contains a Forwarding Header with a Forwarder Link Address that matches its own data link address, the XNP message is retransmitted (via Intranet Relay) to the Destination Link Address in the Forwarding Header.

The Forwarding Header is shown in Table E-1. It consists of the Source Link Address which identifies the originator, the Forwarder Link Address which designates the data link address of the station requested by the joiner to forward the XNP messages, and the Destination Link Address which identifies the final destination.

TABLE E-1. Forwarding header.

OCTET	FIELD IDENTIFICATION	VALUE
1	Message/Block Number: Identifies this as Forwarding Header.	0
2	Source Link Address: Identifies the originator of the XNP message.	1, 2, 4-95
3	Forwarder Link Address: Station selected to forward XNP messages from/to joining station.	4-95
4	Destination Link Address: Identifies the intended recipient of the XNP.	1, 2, 4-95, 127

E.4.1.2 Message and Data Block Structure. XNP messages and data blocks each have the structure shown in Table E-2. Each message and data block starts with a one octet identifier (message or block number) and a one octet length field. These are followed by η data fields. Some data fields consist of multiple octets.

XNP messages are listed in Table E-3. Each of these messages may be combined with one or more of the XNP Data Blocks listed in Table E-4, depending upon the level of detailed information required.

A Terminator Block (Table E-5) designates the end of the XNP message and all associated data blocks. The Terminator Block is required at the end of all XNP messages. Any blocks or messages appearing after the Terminator Block shall be ignored.

TABLE E-2. Message/block structure.

Octet Number	Field Identification
1	identifier octet (message/block number)
2	message/block length
3	data field 1
•	•
•	•
•	•
η+2	data field η

TABLE E-3. XNP messages.

MESSAGE NUMBER	TITLE	DESCRIPTION
20	Join Request	Requests operating parameters assignment, validation, or both.
21	Join Accept	Accepts the Join Request. Provides update of parameters
22	Join Reject	Rejects the Join Request with errors indicated
23	Hello	Announces that a station is entering the network.
24	Goodbye	Announces that station is leaving the network.
25	Parameter Update Request	Requests update of network access and other operational parameters.
26	Parameter Update	Provides update of parameters.
27	Delay Time	Announces a known delay. Provides timer information.

TABLE E-4. XNP data blocks.

BLOCK NUMBER	TITLE	DESCRIPTION
1	Station Identification	Provides a network wide unique identifier for a joiner.
2	Basic Network Parameters	Provides a list of network parameters
3	Hardware Parameters	Provides a list of the hardware parameters associated with reported station.
4	Type 3 Parameters	Provides Type 3 parameters to allow computation of RHD_i and TP.
5	Deterministic NAD Parameters	Provides listing of DAP-NAD and P-NAD parameters to allow computation of access slots.
6	Probabilistic NAD Parameters	Provides listing of R-NAD and H-NAD parameters to allow computation of access slots.
7	RE-NAD Parameters	Provides listing of RE-NAD parameters.
8	Wait Time	Notifies recipient of the amount of time to wait before making required updates.
9	Type 2 Parameters	Provides a list of Type 2 capabilities
10	Type 4 Parameters	Provides a list of Type 4 capabilities
11	NAD Ranking	System ranking for use in deterministic NAD computations.
12	Intranet Parameters	List of parameters for maintaining intranetworking.
13	Error	List of unacceptable parameters.
14-254	undefined	
255	Terminator Block	Notification of end of block transmissions.

TABLE E-5. Terminator block.

OCTET	FIELD IDENTIFICATION	VALUE
1	End of message designator: Identifies the end of the XNP message and associated data blocks.	255

E.4.2 XNP Message Formats. Six messages are used in the procedure for a station to join a network or to request or verify the network operating parameters. These are the Join Request, Join Accept, Join Reject, Parameter Update Request, Parameter Update and Delay Time messages. The Hello message allows an initiating station to announce that it is entering the network. The Goodbye message is issued to report that a station is leaving the network. These messages may be combined with one or more data blocks to provide detailed network operating parameters. These messages are described in the following paragraphs and are depicted in Tables E-6 through E-13.

E.4.2.1 Join Request. The Join Request message (Table E-6) is sent by a station attempting to join a network. The joiner is expected to provide a unique identifier and indicate all implemented capabilities in order to lessen the probability of rejection by the network controller. The unique identifier is used to resolve ambiguities during the joining procedure. The unique identifier is the 24-bit Unit Reference Number (URN) with zeros in the eight most significant bits. If there is no URN a unique identifier must be assigned to each potential user by a mechanism outside the scope of this appendix.

TABLE E-6. Join Request message.

OCTET	FIELD IDENTIFICATION	VALUE
1	Message Number: Identifies specific message content.	20
2-5	Station Identifier: Identifies the station trying to join the network	Unique identifier for the station

The joining station should include data block 2 and, if hardware parameters are known, data block 3. The joining station fills in the applicable data blocks with the parameters supported by the joiner.

Some fields within the Join Request data blocks allow the joiner to set all bits, indicating that all capabilities of that field are supported by the joining station, and the network controller is expected to provide the desired network operating parameters.

The Join Request message with no data blocks will suffice provided joiner has every capability listed in data block 2, has not been pre-assigned a data link address, is capable of all optional data link layer service types, and does not know hardware parameters.

E.4.2.2 Join Accept. The Join Accept message (Table E-7) is sent by a network controller in response to the Join Request message, provided entry to the network has been approved. If there is no network controller, any station may send a Join Accept message in response to the Join Request message. Actual network operating parameters will be provided in the appropriate data blocks combined with the Join Accept message. The appropriate data blocks appended to the Join Accept message depend upon the network configuration and the capabilities of the joining station. It will typically include data blocks 2, 4, 9 (if joiner indicated a Type 2 capability and there exists a network default), 10 (if joiner indicated a Type 4 capability) and either block 5, 6, or 7 (depending upon network access method in use).

The Join Accept message may include data block 8 to specify a period that the joining station should wait after sending a Hello message before it can assume its selected data link address has been accepted.

TABLE E-7. Join Accept message.

OCTET	FIELD IDENTIFICATION	VALUE
1	Message Number: Identifies specific message content.	21
2-5	Station Identifier: Identifies the station trying to join the network	Unique identifier for the station
6-17	Address Map: Bit map of addresses	Bits correspond to data link addresses 0 through 95. Bit 0 corresponds to data link address 0. A bit set to one specifies the address is not available (i.e., it is already in use).
18	Communications Functions: An indication of the network type (centralized or distributed) in use.	0 = No Net Controller 1 = Centralized 2 = Distributed

E.4.2.3 Join Reject. The Join Reject message (Table E-8) is sent by a network controller when entry to the network has not been approved. The Join Reject should be interpreted as being applied to the station identified in the Station Identifier field. Join Reject messages originated by any station other than a network controller should be discarded and ignored.

The Join Reject message is sent in response to the Join Request message when the reason for rejection is that the parameters provided are not presently acceptable in this network. An error indication is provided with the Join Reject to clearly identify the unacceptable parameter(s). This error indication may be data block 13, which lists the unacceptable parameters and/or one or more other data blocks correcting the unacceptable parameter(s).

The Join Reject message is also sent in response to a Hello message to indicate that the joiner has selected a data link address that is in use. Rejection of a joining station's use of a data link address can be accomplished with only the basic Join Reject information fields, no data blocks (except the Termination Block) are required. Unless the joining station can correct the error(s), entry via XNP is not possible.

When a station receives a Join Reject message, the station identified in the Station Identifier field shall be removed from its topology tables unless it is a static node (link quality is 7).

TABLE E-8. Join Reject message.

OCTET	FIELD IDENTIFICATION	VALUE
1	Message Number: Identifies specific message content.	22
2-5	Station Identifier: Identifies the station trying to join the network	Unique identifier for the station
6	Rejected Link Address. The data link address being rejected because it is already in use. Joining station must select another address.	Rejected data link address. 0 = Rejection is for a reason other than data link address.
7-18	Address Map: Bit map of addresses.	Bits correspond to data link addresses 0 through 95. Bit 0 corresponds to data link address 0. A bit set to one specifies the address is not available.
19	Communications Functions: An indication of the network type (centralized or distributed) in use.	0 = unknown 1 = Centralized 2 = Distributed

E.4.2.4 Hello Message. The Hello message (Table E-9) is sent by a station after the network operating parameters are known and the station is ready to enter the network. The message contains the data link address of the station entering the network. Address tables within receiving stations are updated, if necessary, with the new address information. When a station receives a Hello message, it shall update its topology tables.

TABLE E-9. Hello message.

OCTET	FIELD IDENTIFICATION	VALUE
1	Message Number: Identifies specific message content.	23
2-5	Station Identifier: Identifies the station trying to join the network	Unique identifier for the station
6	Selected Link Address: The actual data link address selected for use by this station.	Data link address selected by this station.

E.4.2.5 Goodbye Message. The Goodbye message (Table E-10) is sent by a station to notify the network controller and other network subscribers that it is leaving the network. The data link address used by the receiving station is made available for re-use by another station. Address tables within the receiving stations are updated, if necessary.

Before a station sends a Goodbye message, it should disconnect all Type 2 connections and broadcast a URNR and DRNR to indicate it will no longer receive frames. When a station receives a Goodbye message, it shall update its topology tables.

TABLE E-10. Goodbye message.

OCTET	FIELD IDENTIFICATION	VALUE
1	Message Number: Identifies specific message content.	24
2-5	Station Identifier: Identifies the station leaving the network.	Unique identifier for this station
6	Released Link Address: The data link address of the station leaving the network.	Data link address of this station.

E.4.2.6 Parameter Update Request Message. A station that is out of operation for some period of time or experiences a system failure may be unaware of changes to the network operating procedures/parameters or may have lost all record of the operating parameters. Once the outage or failure is corrected, the station may send this Parameter Update Request message (Table E-11) to obtain new/changed parameters. The station may use this message to obtain an update of any or all parameters by attaching data blocks identifying the parameters that need to be updated.

TABLE E-11. Parameter Update Request message.

OCTET	FIELD IDENTIFICATION	VALUE
1	Message Number: Identifies specific message content.	25
2-5	Station Identifier: Identifies this station.	Unique identifier for the station
6	Update Requested: Update of parameters for requesting station or for all stations on the network. Actual parameters requested shall be stated in attached data blocks.	Bits set to one designate the following: 0 = requesting station 1 = all stations 2 - 7 undefined

E.4.2.7 Parameter Update Message. The Parameter Update message (Table E-12) shall be sent in response to the Parameter Update Request message. It may be sent by the network controller before sending a Join Accept message in response to a Join Request message. The Parameter Update message may include data block 8 to specify the time period, after receipt of the message that the network parameters become effective.

TABLE E-12. Parameter Update message.

OCTET	FIELD IDENTIFICATION	VALUE
1	Message Number: Identifies specific message content.	26
2-5	Station Identifier: Identifies a station.	Unique identifier for the station
6-17	Address Map: Bit map of addresses.	Bits correspond to data link addresses 0 through 95. Bit 0 corresponds to data link address 0. A bit set to one specifies address not available.
18	Communications Functions: An indication of the network type (centralized or distributed) in use.	0 = unknown 1 = Centralized 2 = Distributed

E.4.2.8 Delay Time Message. The Delay Time message (Table E-13) is sent by a Forwarder in response to a broadcast Join Request message. It provides an indication to the joiner of how long a delay should be expected before the Forwarder will return a Join Accept message from the network controller after a Forwarded Join Request message is received from the joining station.

TABLE E-13. Delay Time message.

OCTET	FIELD IDENTIFICATION	VALUE
1	Message Number: Identifies specific message content.	27
2-5	Station Identifier: Identifies the station trying to join the network	Unique identifier for the station
6	Time: The amount of time the joiner should expect to wait for a Join Accept message after sending a Join Request through this Forwarder.	1 to 255 seconds in 1 second increments

E.4.3 XNP Data Block Formats. One or more additional XNP Data Blocks may appear before the Terminator Block in each XNP message to either provide specific network operating parameters or to request specific parameters. The additional XNP Data Blocks are described in the following paragraphs and are depicted in Tables E-14 through E-26.

E.4.3.1 Block 1, Station Identification. Block 1 (Table E-14) consists of one field which is used to identify the station being reported. It is used with the Parameter Update message to identify the station to which the parameters apply, in Block 2, and/or Block 11 (that must be preceded by Block 1).

TABLE E-14. Station ID.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block.	1
2	Length: Indicates the length of the Station ID block in octets	6
3-6	Unique Identifier: Identifies the station trying to join the network or being updated.	Unique identifier for the station

E.4.3.2 Block 2, Basic Network Parameters. This block (Table E-15) is used to define basic network capabilities of a joining station, a requesting station or any other station identified by Block 1. It is mandatory in the Join Request message to identify capabilities of the joining station, unless the joining station has all possible capabilities listed. It is optional with the Join Accept message, Hello message, Parameter Update Request message and Parameter Update message.

TABLE E-15. Basic network parameters.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block.	2
2	Length: Indicates the length of the Basic Network Parameters block in octets	13
3	Link Address: Identifies the data link address of the station.	0 = unknown 4 to 95 = actual data link address
4	Station Class: The types of data link services available (See 5.3.3.5).	0 = Class A 1 = Class B 2 = Class C 3 = Class D
5	NAD Methods: Identifies either the NAD methods available by a station or the specific NAD method being used in a network.	Bit map: 0=R-NAD 1=H-NAD 2=P-NAD 3=DAP-NAD 4=RE-NAD
6-9	Group Address: Bit map that identifies the group address(es) that the station is a member of.	Bit map: LSB = 96 MSB-1 = 126
10	Concatenation Capability: Indicates the types of concatenation supported by the reporting station.	Bit map: 0 = Physical layer 1 = Data link layer
11	EDC/TDC/Scrambling Mode: Bit map which identifies the FEC, TDC and Scrambling capabilities.	Bit Map: 0=Half-rate Golay FEC 1=TDC 2=V.33 Scrambling 3=V.36 Scrambling 4=Robust Comm. Protocol
12-13	Max. UI, DIA and I Info. Octets: Indicates the largest information field size that can be handled by the reporting station or that is allowed on the network.	708 - 3345 octets 65535 = requested

E.4.3.3 Block 3, Hardware Parameters. Hardware parameters defined by Block 3 (Table E-16) are required to enable computation of TP, RHD and Net_Busy_Detect_Time described in Appendix C. Although not mandatory with any message, it could lead to erroneous network control computations resulting in collisions if the information is not provided in a Join Request message.

TABLE E-16. Hardware parameters.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block	3
2	Length: Indicates the length of the Hardware Parameters block in octets	18
3	Equipment Preamble Time (EPRE): Network Access Control parameter defined in Appendix C.	from 0 in 1 msec increments
5-6	Phasing Time: Network Access Control parameter defined in Appendix C.	0 - 10000 in 1 msec increments
7-8	Equipment Lag Time (ELAG): Network Access Control parameter defined in Appendix C.	from 0 in 1 msec increments
9-10	Turnaround Time (TURN): Network Access Control parameter defined in Appendix C.	from 0 in 1 msec increments
11-12	Tolerance Time (TOL): Network Access Control parameter defined in Appendix C.	0 - 500 msec in 1 msec increments
13-14	DTE Processing Time (DTEPROC): Network Access Control parameter defined in Appendix C.	from 0 in 1 msec increments
15	DTE Acknowledgment Time (DTEACK): Network Access Control parameter defined in Appendix C.	0 - 254 in 1 msec increments
16-17	Net Busy Detect Time, B: The time to detect data network busy.	from 0 in 1 msec increments
18	Mode Of Operation. Identifies the Physical Layer protocol capabilities of the station or being used in the network. Multiple bits may be set.	Bit Map: 0=Synchronous Mode 1=Asynchronous Mode 2=Packet Mode 3=Robust Comm. Protocol

E.4.3.4 Block 4, Type 3 Parameters. These parameters (Table E-17) are required for data link Type 3 (acknowledged Type 1) operations and are mandatory with the Join Accept message to provide the joining station with sufficient information to use Type 3 in the network. This block is optional with the Parameter Update Request message and the Parameter Update message.

TABLE E-17. Type 3 parameters.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block.	4
2	Length: Indicates the length of the Type 3 Parameters block in octets	3
3	Type 3 Retransmissions: The maximum number of times to retransmit an unacknowledged frame.	0 to 5

E.4.3.5 Block 5, Deterministic NAD Parameters. This block (Table E-18) defines parameters needed to allow operation in a network configured for deterministic network access (DAP-NAD or P-NAD) operations. It is mandatory with the Join Accept message if the network being joined is operating with P-NAD or DAP-NAD. It may also be used with the Parameter Update Request message and the Parameter Update message. This block is required in the Parameter Update message if it is being used to announce the network's access procedures are changing to either P-NAD or DAP-NAD.

TABLE E-18. Deterministic NAD parameters.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block	5
2	Length: Indicates the length of the Deterministic NAD Parameters block in octets	6
3	Number Of Stations: Indicates the number of stations participating on the network. Used in NAD calculations.	2 - 95
4	Number Of NAD Priorities: Number of priorities to be considered in P-NAD and DAP-NAD method.	1 - 8
5	Number Of NAD Slots: Indicates the number of NAD slots available for P-NAD and DAP-NAD operations.	1 - 127
6	NAD Slot Duration: Duration of the NAD time slot for NAD operations.	0 - 2540 msec in 10 msec increments

E.4.3.6 Block 6, Probabilistic NAD Parameters. Block 6 (Table E-19) provides network access delay operating parameters for probabilistic networks (R-NAD or H-NAD). It is mandatory with the Join Accept message to provide the joining station with required operating parameters if the network is configured for either R-NAD or H-NAD. It is optional with the Parameter Update Request message and the Parameter Update message. This block is required in the Parameter Update message if it is being used to announce the network's access procedures are changing to either R-NAD or H-NAD.

TABLE E-19. Probabilistic NAD parameters.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block	6
2	Length: Indicates the length of the Probabilistic NAD Parameters block in octets	7
3	Number Of Stations: Indicates the number of stations participating on the network. Used in NAD calculations.	2 - 95 stations on the network
4	Number Of NAD Priorities: Number of priorities to be considered in R-NAD and H-NAD method.	1 - 8
5	Urgent Percent: The percentage of urgent (%U) frames expected in an average 24-hour period. Used in the H-NAD calculation.	0 - 100% This value plus Priority Percent value must be less than or equal to 100%
6	Priority Percent: The percentage of priority (%P) frames expected in an average 24-hour period. Used in the H-NAD calculation.	0 - 100% This value plus Urgent Percent value must be less than or equal to 100%
7	Traffic Load: The amount of network traffic expected. Used in the H-NAD calculation.	0=normal, 1=heavy, 2=light

E.4.3.7 Block 7, RE-NAD Parameters. These parameters (Table E-20) are required for stations in a network operating with RE-NAD. It is mandatory with the Join Accept message to provide joining stations with network access parameters if the network being joined is configured for RE-NAD. It is optional with the Parameter Update Request message and the Parameter Update message. This block is required in the Parameter Update message if it is being used to announce the network's access procedures are changing to RE-NAD.

TABLE E-20. RE-NAD parameters.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block.	7
2	Length: Indicates the length of the RE-NAD Parameters block in octets	TBD
3-	undefined	

E.4.3.8 Block 8, Wait Time. This block (Table E-21) is used with the Join Accept message and Parameter Update message to specify a delay. When used with the Join Accept message, it indicates how long the Joining station should wait after sending a Hello message before it can assume its entry to the network is accepted. When used with the Parameter Update message, it indicates when new operating parameters become effective.

TABLE E-21. Wait time.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block	8
2	Length: Indicates the length of the Wait Time block in octets	3
3	Wait Time: Delay period.	1 to 255 seconds in 1 second increments

E.4.3.9 Block 9, Type 2 Parameters. This block (Table E-22) identifies individual or network operating parameters for stations capable of optional Type 2 operations. It may be used with the Join Accept message, Parameter Update Request message and Parameter Update message.

E.4.3.10 Block 10, Type 4 Parameters. Type 4 parameters (Table E-23) are required for stations in a network which are capable of Type 4 operations. It may be used with the Join Accept message, Parameter Update Request message and Parameter Update message.

E.4.3.11 Block 11, NAD Ranking. This block (Table E-24) provides ranking of a station in a deterministic network access configured network. It is mandatory if the network is configured for either P-NAD or DAP-NAD. It may be used with the Join Accept message or the Parameter Update message. In the Parameter Update message, it may be repeated to identify ranking of each station in the network. In this case, this block will appear once for each station on the network and will be preceded by block 1 to identify the station to which the ranking applies.

E.4.3.12 Block 12, Intranet Parameters. The Intranet parameters (Table E-25) must be provided to joining stations to provide information for Intranet relaying within the local network. This block shall be included with the Join Accept and Parameter Update messages.

E.4.3.13 Block 13, Error. Block 13 (Table E-26) is encoded in Block/Byte number pairs indicating the starting byte number of the field containing the error.

Block 13 may be included with the Join Reject message to indicate the reasons that a Join Request is being rejected.

TABLE E-22. Type 2 parameters.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block.	9
2	Length: Indicates the length of the Type 2 Parameters block in octets	12
3-4	ACK Timer: The amount of time before Waiting Acknowledgment procedures are initiated.	10 - 1800 seconds in 1 second increments.
5	P-Bit Timer: The amount of time before Waiting Acknowledgment procedures are initiated when P-bit was set to 1.	10-60 seconds in 1 second increments.
6-7	Reject Timer: The amount of time before re-sending the REJ or SREJ if no response is received.	20 - 3600 seconds in 1 second increments.
8	Max. Transmissions, N2: The maximum number of times an I frame may be transmitted.	0 - 5
9	K Window: The maximum number of outstanding I PDUs allowed on a connection.	1 - 127
10	K2 Threshold: The maximum number of unacknowledged I PDUs on a connection before an acknowledgment is requested.	1 - 127
11	K3 Threshold: The maximum number of unacknowledged I PDUs on a connection before an acknowledgment must be sent.	1 - 127
12	Response Delay Timer: The amount of time that a station waits after an I PDU with its P-bit set to 0 is received before sending an acknowledgment.	1 - 1800 seconds in 1 second increments.

TABLE E-23. Type 4 parameters.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block	10
2	Length: Indicates the length of the Type 4 Parameters block in octets	6
3-4	ACK Timer: The amount of time before a DIA is retransmitted.	50 - 1200 tenths of seconds
5	K Window: The maximum number of outstanding DIA frames allowed for a station.	5 - 20
6	Max. Transmissions: The maximum number of times a DIA frame may be transmitted.	0 - 5

TABLE E-24. NAD ranking.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block	11
2	Length: Indicates the length of the NAD Ranking Parameters block in octets	3
3	Subscriber Rank: Identifies the ranking of this station relative to other stations on the network. Used in P-NAD and DAP-NAD calculations to determine the actual order of network access.	1-127 with 1 being highest.

TABLE E-25. Intranet parameters.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block	12
2	Length: Indicates the length of the Intranet Parameters block in octets	11
3	Min Update Per: Topology updates should not be transmitted more often than once every Min_Update_Per.	1 minute increments 0=No Updates
4	Topology Update Precedence: The precedence of Topology Update messages.	0 = Routine 1 = Priority 2 = Immediate 3 = Flash 4 = Flash Override 5 = CRITIC/ECP 6 = Internet Control 7 = Network Control 8 - 255 = undefined
5	Relayer Status: Indicates if the station is a relayer or non-relayer.	0=No Relay 1=Relay
6-7	ACK Timer (fixed factor): The base time to wait before retransmitting an unacknowledged Intranet message.	0 to 600 in seconds
8-9	ACK Timer (proportional factor): The amount of time to add to the fixed factor for each hop to the furthest destination of an Intranet message.	0 to 600 in seconds
10	Retransmit Count: The maximum number of retransmissions of an Intranet message.	1 to 4
11	Link Failure Threshold: The number of data link acknowledgment failures required to change a station's status to failed.	1 to 7

TABLE E-26. Error.

OCTET	FIELD IDENTIFICATION	VALUE
1	Block Number: Identifies specific data block	13
2	Length: Indicates the length of the Error block in octets	4 + 2n, where n = the number of errors
3	Message/Block Number 1: Indicates the message or block containing the first error.	1 through 12, 20 through 27
4	Byte Number 1: Indicates the first octet of the field within the message or block that contains the first error.	1 through 255
	• • •	
3 + 2n	Message/Block Number n: Indicates the message or block containing the nth error.	1 through 13, 20 through 27
4 + 2n	Byte Number n: Indicates the first octet of the field within the message or block that contains the nth error.	1 through 255

E.5. XNP Message Exchange. XNP messages shall be exchanged using a UI command frame as shown in Figure E-3.

FLAG	Source Address	Destination Address	Control Field	Intranet Header	XNP Information	FCS	FLAG

FIGURE E-3. UI frame containing XNP message.

E.5.1 Data Link Addressing. Data link address 1 is a special address for a station to use while joining the network if it has not been pre-assigned a data link address. If a station has not been assigned a data link address, it shall use this special data link address for network entry until an individual data link address has been assigned or selected. Since multiple stations may be attempting to join the network at the same time, the Station Identifier field in each XNP message is used to uniquely identify the station.

Data link address 2 is a special address reserved for the network control station. Joining stations, forwarders and relayers use the special address 2 to address the network control station. The forwarder shall provide the full source directed relay path to the network controller at the Intranet layer. The network controller shall use this same path in reverse to reach the joining station through the forwarder. The Station Identifier field in the XNP messages used to uniquely identify stations during the joining process.

In a network using distributed control there may be more than one network controller. Network controllers in a network using distributed control may use their individual address at the data link layer of the UI frame carrying the Join Accept message. Also, in networks with no network controller, any station may respond to a Join Request message with a Join Accept message using their own individual address.

E.5.2 Poll/Final Bit. Use of UI poll/final bits is allowed but not recommended for use with XNP Join Request, Join Accept and Join Reject messages because network timing parameters for Type 1 final-bit responses are either unknown or subject to change during the network joining process.

E.5.3 Network Access. MIL-STD-188-220 allows a network to choose among the network access delay methods defined in Appendix C. Each station that operates on the network must use the same method. If the station does not know this information before joining the network, the Join Request message allows a station to learn the network access method. In the case that the network access method is unknown, a random method (R-NAD or RE-NAD) shall be used for the Join Request message. When R-NAD is used, the default number of stations shall be 7 unless another number is known.

E.6. Network Joining Procedures. Joining procedures depend upon the network configuration prevailing at the time of attempted entry. If the network is operating with a centralized network controller, operating parameters will be provided only by the network controller. If operating with distributed network control, any existing network controller is capable of providing operating parameters. The joining station may or may not be aware of the network configuration.

E.6.1 Joining Concept. In general, the basic network joining procedure depicted in Figure E-4 is followed. The joining station sends a Join Request message that contains its MIL-STD-188-220 capabilities and unique identifier. The responding network controller compares the joiner's capabilities with current network operating parameters. If an error is found which precludes acceptance into the network, the network controller returns a Join Reject message to the joiner. The Join Reject message may include all of the correct parameters in appropriate data blocks of the message and/or use the error message to identify errors. If the errors can be corrected by the joiner, a new Join Request message with corrected parameters can be sent. If the errors are not correctable, automatic joining using XNP is not possible.

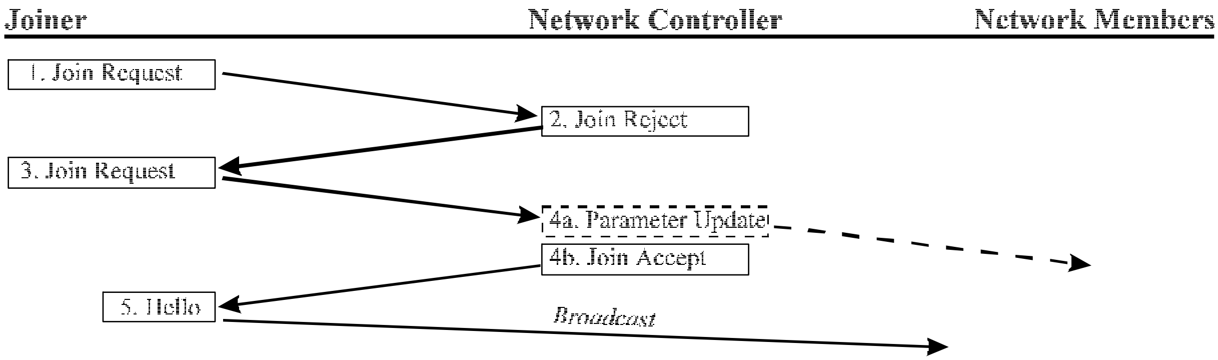


FIGURE E-4. Joining concept.

If there are no errors in the parameters contained in the Join Request message, a Join Accept message is sent by the network controller after entering the parameters for any empty or updated parameter fields. The network controller may have to update the data fields filled in by the joiner since it is possible that the joining station has capabilities above and beyond those being used within the network. If adding the joiner to the network will cause a change to the network operating parameters (e.g., number of stations), the network controller may announce the new network parameters with the Parameter Update message.

The Join Accept message contains an address bit map identifying data link addresses that can be selected by the joining station. When the joining station receives a Join Accept message response from the network controller, it shall select a data link address from the address bit map and broadcast a Hello message announcing entry to the network. Other members of the network shall update their topology tables upon receipt of the Hello message.

A network controller may send a Join Reject to remove any station from the network at any time. Since the Join Reject may be sent to prevent a joining station from selecting an already used address, the Join Reject message should be interpreted as being applied to the station identified in the Station Identifier field of the message. Other network members of the network shall update their topology tables upon receipt of the Join Reject message.

When a station leaves a network, it shall send a Goodbye message to announce that the data link address is available for use by another station. Other members of the network shall update their topology tables upon receipt of the Goodbye message.

E.6.2 Procedures for Joining a Network with Centralized Network Control. The procedure for joining a network with centralized network control is depicted in Figure E-5. To simplify the discussion and the figure, Join Reject and Parameter Update messages discussed in the basic Joining Concept are not included.

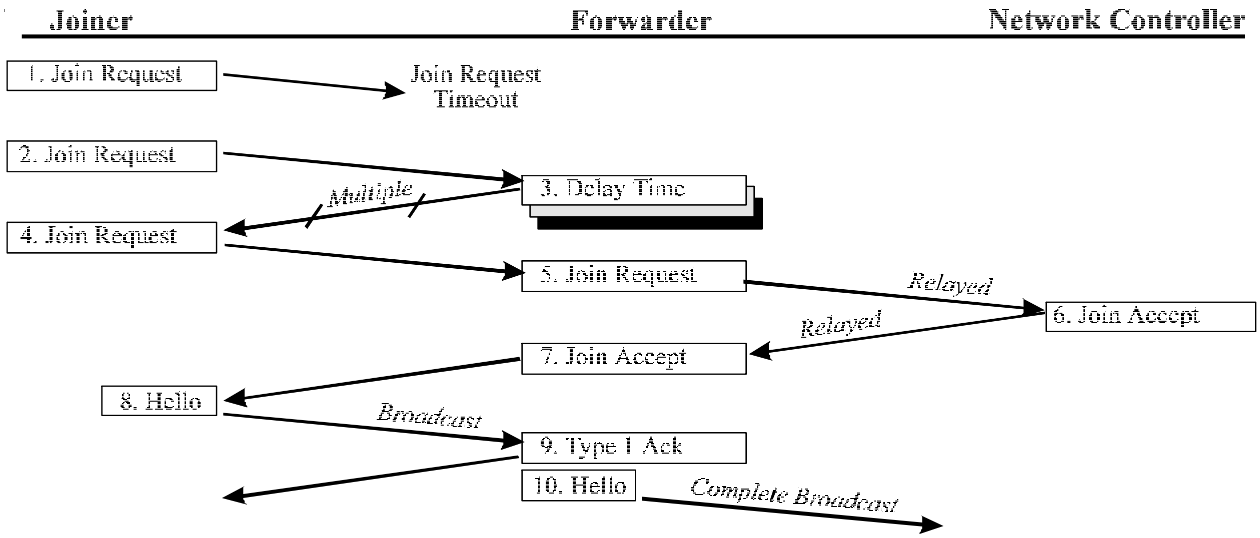


FIGURE E-5. Joining a centralized network.

The joining station shall send a Join Request message to the network controller. The Join Request message shall be addressed to the network controller using the special data link address of 2 as the destination and the special data link address of 1 as the source in the UI frame. If the joining station is unable to contact the network controller because of distance or topology, there will be no response to the Join Request message. In this event, the joining station shall retransmit the Join Request message after the Join Request interval timer expires until the Maximum Number of Join Retries has been exceeded or until either a Join Reject or Join Accept message is received.

If the maximum number of Join Retries is exceeded, the joining station shall then address a UI frame containing the Join Request message to the Global address. The joining station shall continue sending the Join Request message to the Global address after the Join Request interval expires until a response is received from an existing network member.

All network members that receive the globally addressed Join Request message, and intend to participate in the joining procedure, shall send a Delay Time message with an XNP Forwarding Header in response to the joining station. The joining station shall select one of the responding stations as forwarder and resend the Join Request to the network controller using the forwarding parameters in the Forwarding Header received from the selected station. The selected forwarder shall relay this Join Request to the network controller and forward the network controller's response (Join Accept or Join Reject message) back to the joining station. The Join Accept message shall specify a list of unused data link addresses.

The joining station shall expect the network controller response before expiration of the Delay Timer (the period of time specified in the selected forwarder's Delay Time message). If the Delay Timer expires, the joining station shall try each responder in turn in an attempt to contact the network controller.

When the joining station receives a Join Accept message response from the network controller, it shall prepare a Hello message announcing entry to the network. The Hello message shall use the joining station's assigned individual address (selected from the Join Accept's list of unused data link addresses) as the source address and shall include both the forwarder's individual address and the Global multicast address as destinations in the UI frame. The UI frame carrying this Hello message shall have the P-bit set.

The forwarder shall return a Type 1 acknowledgment to the joining station and then complete the broadcast of the Hello message to all network members. This complete broadcast involves relaying the Hello message, including Forwarding Header, using appropriate Intranet procedures (e.g., Source Directed Relay). The forwarder shall set the maximum hop count in the Intranet Header of the message to restrict the amount of relaying.

E.6.3 Procedures for Joining a Network with Distributed Network Control. The procedure for joining a network with distributed network control is depicted in Figure E-6. To simplify the discussion and the figure, Join Reject and Parameter Update messages discussed in the basic Joining Concept are not included.

The joining station shall send a Join Request message to a network controller using the special data link address of 2 as the destination and the special data link address of 1 as the source in the UI frame. One or more existing network members operating as network controller shall respond with a Join Accept message or a Join Reject message. Each Join Accept message shall specify a list of unused data link addresses and a Wait Time.

If the joining station is unable to contact a network controller because of distance or topology, there will be no response to the Join Request message. The joining station shall retransmit the Join Request message after the Join Request interval timer expires until the Maximum Number of Join Retries has been exceeded or until either a Join Reject or Join Accept message is received.

If the maximum Number of Join Retries is exceeded, the joining station shall then address a UI frame containing the Join Request message to the Global address. The joining station shall continue sending the Join Request message to the Global address after the Join Request interval expires until a response is received from an existing network member.

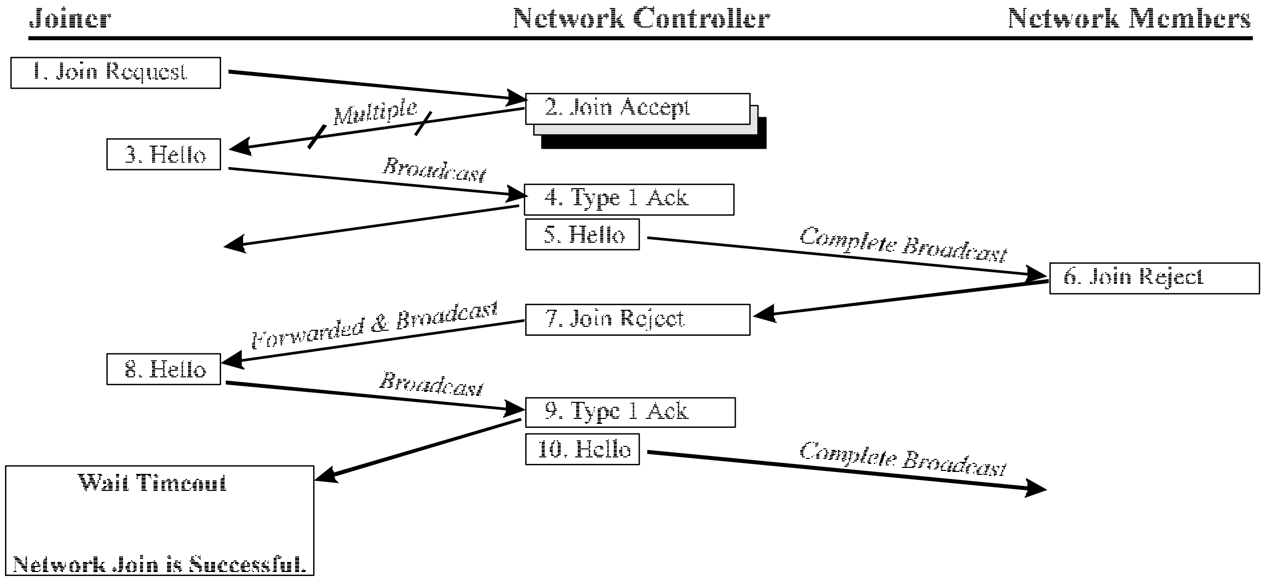


FIGURE E-6. Joining a distributed network.

All network members that receive the globally addressed Join Request, and intend to participate in the joining procedure, shall send a Delay Time message with an XNP Forwarding Header in response to the joining station. The joining station shall select one of the responding stations as forwarder and resend the Join Request to a network controller using the forwarding parameters in the Forwarding Header received from the selected station. The selected forwarder shall relay this Join Request to the network controller and forward the network controller's response (Join Accept or Join Reject message) back to the joining station.

When the joining station receives the Join Accept message responses from the network controllers (via forwarding if necessary), it shall prepare a Hello message announcing entry to the network. The Hello message shall use the joining station's individual address (selected from the Join Accept's list of unused data link addresses) as the source address and shall use the Global multicast address and the selected network controller or forwarder as destinations in the UI frame. The UI frame carrying this Hello message shall have the P-bit set.

The selected network controller or forwarder shall return a Type 1 acknowledgment to the joining station and then complete the broadcast of the Hello message to all network members. This complete broadcast involves relaying the Hello message, including Forwarding Header, using appropriate Intranet procedures (e.g., Source Directed Relay). The selected forwarder shall set the maximum hop count in the Intranet Header of the message to restrict the amount of relaying.

The joining station shall start a Wait Timer upon receipt of the Type 1 acknowledgment from the forwarder. If the Wait Timer expires, the selected individual address may be used by the Joiner in the network.

If a Join Reject response is received before expiration of the Wait Timer, it indicates the selected address is already in use and the joining station shall select another data link address and send another Hello message. A Join Reject message is sent by the rejecting network controller to the forwarder. The forwarder forwards the Join Reject message to the joiner using the individual address selected by the joiner and also completes the broadcast of the Join Reject to all network members. This complete broadcast involves relaying the Join Reject message, including Forwarding Header, using appropriate Intranet procedures (e.g., Source Directed Relay). The forwarding network controller shall set the maximum hop count in the Intranet Header of the message to restrict the amount of relaying.

E.6.4. Joining Procedure Examples.

E.6.4.1 Centralized Network Control, Fully Connected Network. In this example, there is a single, centralized network controller and it is in direct line of sight to the joiner. The network is using data link Type 1 only and is using DAP-NAD. The joining station has all optional capabilities. Therefore the sequence of events is shown in Figure E-7 and is described in section E.6.4.1.1. Detailed message formats are provided in section E.6.4.1.2.

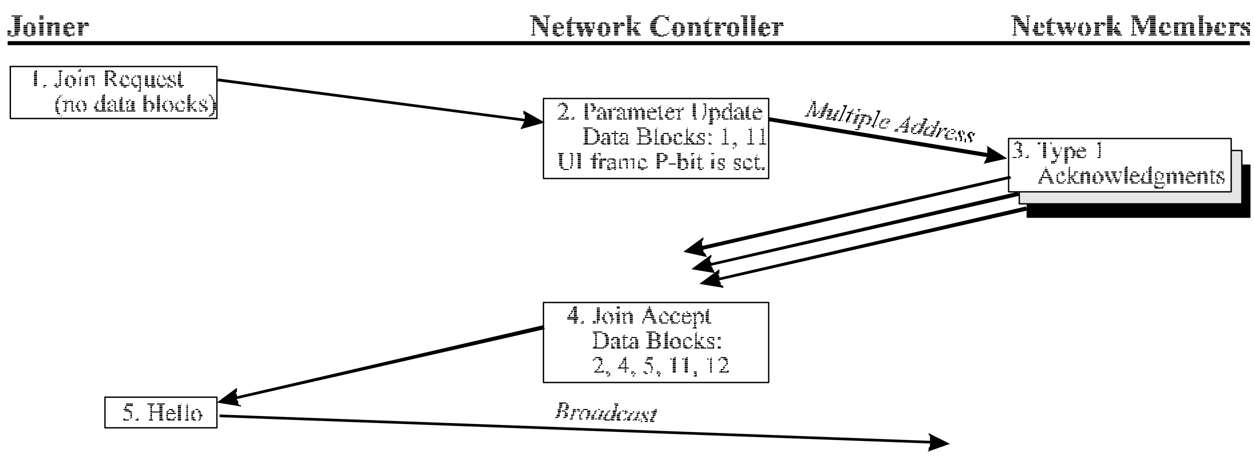


FIGURE E-7. Joining a fully connected, centralized network.

E.6.4.1.1 Sequence of Events.

1. The joining station sends a UI frame with a Join Request message to the network controller requesting entry to the network. No data blocks are appended since the joining station does not have knowledge of hardware parameters, but does have all optional capabilities.

2. The network controller computes the ranking for DAP-NAD and transmits a Parameter Update message to all network members. This Parameter Update

message includes blocks 1 and 11 to designate the order of NAD access for all stations in the network. It is sent with the P-bit set to 1 to provide some level of assurance that it has been received and implemented by all participants.

3. Each network participant sends a Type 1 Acknowledgment of the UI frame carrying the Parameter Update message to the network controller.

4. The network controller responds with a Join Accept message to the joiner with the Communications Functions field set to Centralized and only one bit set to 0 in the Address Map to specify the address assigned to the joining station. Data block 2, block 4, block 5, and block 12 are appended to the Join Accept message to provide the network operating parameters to the joining station. The Join Accept also contains Blocks 1 and 11 to provide the relative NAD rankings for each network member.

5. The joining station sends a Hello message to announce its entry into the network.

E.6.4.1.2 Message Formats.

1. Join Request Message

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	000	(Initial)
T-bits	00	(No Info)
Link Layer Header (bits, LSB on right):		
Source Address	00000010	(Net Entry)
Destination Address	00000101	(network controller)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Message Number	20	(Join Request)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

2. Parameter Update Message

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	000	(Initial)
T-bits	01	(DAP-NAD)
Data Link Precedence	00	(Urgent)
First Subscriber Number	000010	(network controller)
Link Layer Header (bits, LSB on right):		
Source Address	00000100	(network controller)
Destination Address(es)	[up to 16 data link addresses]	
Control Field	00010011	(UI, ACK required)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Message Number	26	(Parameter Update)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
XNP Data Blocks		
Data Block 1	Station Identification (station #1)	
Data Block 11	NAD Ranking (station #1)	
• • •		
Data Block 1	Station Ident. (last station)	
Data Block 11	NAD Ranking (last station)	
Terminator Block	255	

3. Type 1 Acknowledgment

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	yyy	(each station's current #)
T-bits	01	(DAP-NAD)
Data Link Precedence	00	(Urgent)
First Subscriber Number	zzzzzz	(using old ranking)
Link Layer Header (bits, LSB on right):		
Source Address	0aaaaaa1	(Acknowledging station)
Destination Address	00000101	(network controller)
Control Field	00110011	(URR response)

4. Join Accept Message

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	000	(Initial)
T-bits	01	(DAP-NAD)
Data Link Precedence	01	(Priority)
First Subscriber Number	xxxxxx	(Joiner)
Link Layer Header (bits, LSB on right):		
Source Address	00000100	(network controller)
Destination Address	00000011	(Net Entry)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Message Number	21	(Join Accept)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Address Map	[one bit set to 0, specifying the joiner's link address	
Comm. Functions	1	(Centralized)
XNP Data Blocks:		
Data Block 2	Basic Network Parameters	
Data Block 4	Type 3 Network Parameters	
Data Block 5	Deterministic NAD Parameters	
Data Block 12	Intranet Parameters	
Data Block 1	Station Identification (station #1)	
Data Block 11	NAD Ranking (station #1)	
•		
•		
•		
Data Block 1	Station Ident. (last station)	
Data Block 11	NAD Ranking (last station)	
Terminator Block	255	

5. Hello Message

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	000	(Initial)
T-bits	01	(DAP-NAD)
Data Link Precedence	00	(Urgent)
First Subscriber Number	000010	(network controller)
Link Layer Header (bits, LSB on right):		
Source Address	0xxxxxx0	(Joiner)
Destination Address	11111111	(Global Multicast)
Control Field	11000000	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	Minimum)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Message Number	23	(Hello)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

E.6.4.2 Centralized Network Control, Disconnected Joiner. In this example, there is a single, centralized network controller and it is not in direct line of sight to the joiner. The network is using data link Types 1 and 2, but not Type 4, and is using DAP-NAD. The joining station has all optional capabilities. Therefore the sequence of events is shown in Figure E-8 and is described in section E.6.4.2.1. Detailed message formats are provided in section E.6.4.2.2.

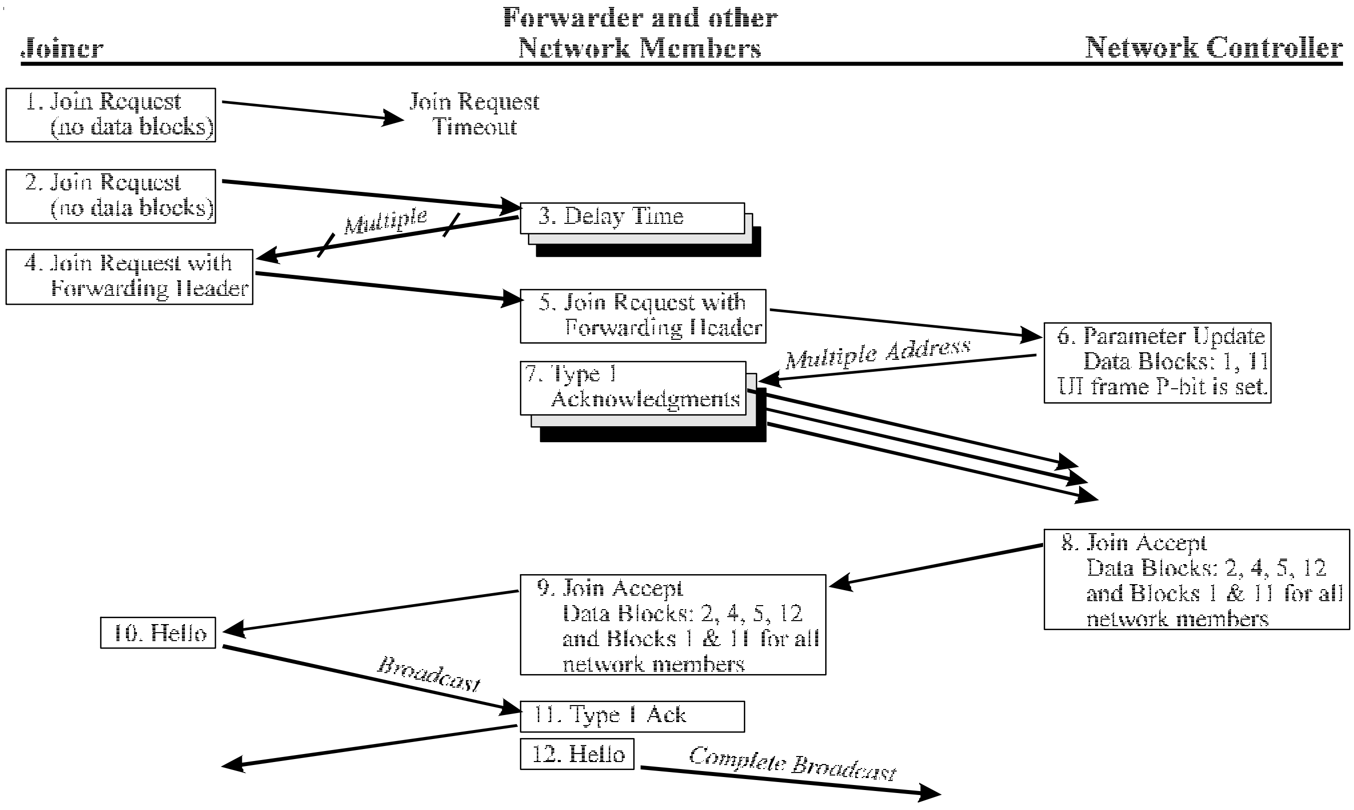


FIGURE E-8. Joining a disconnected, centralized network.

E.6.4.2.1 Sequence of Events.

1. The joining station sends a UI frame with a Join Request message to the network controller requesting entry to the network. No data blocks are appended since the joining station does not have knowledge of hardware parameters, but does have all optional capabilities.

   Because the joiner is not in direct line of sight with the network controller, network controller does not receive the Join Request and there is no response.

2. The joining station sends a UI Command with a Join Request message to the Global Multicast data link address requesting entry to the network. This Join Request message has a Forwarding Header identifying the network controller as the Destination.

3. The Join Request message is received by stations 44, 25 and 31. These three stations send a Delay Time message to the joining station.

4. The joining station selects station 25 as the forwarder and uses this station to forward a Join Request message to the network controller.

5. Station 25 forwards the Join Request message to the network controller for the joining station.

6. The network controller computes the ranking for DAP-NAD and transmits a Parameter Update message to all network members. This Parameter Update message includes blocks 1 and 11 to designate the order of NAD access for all stations in the network. It is sent with the P-bit set to 1 to provide some level of assurance that it has been received and implemented by all participants.

   All stations update to new subscriber order.

7. Each network participant sends a Type 1 Acknowledgment of the UI frame carrying the Parameter Update message to the network controller.

8. The network controller responds with a Join Accept message to the joiner with the Communications Functions field set to Centralized and only one bit set to 0 in the Address Map to specify the address assigned to the joining station. Data block 2, block 4, block 5, block 9 and block 12 are appended to the Join Accept message to provide the network operating parameters to the joining station. The Join Accept also contains Blocks 1 and 11 to provide the relative NAD rankings for each network member.

9. Station 25 forwards network controller's Join Accept message (and all data blocks) to the joining station.

10. The joining station sends a Hello message to announce its entry into the network. This Hello message is broadcast locally, and also addressed to forwarding station 25 so that it can be broadcast completely through the network.

11. Forwarding station 25 sends a Type 1 Acknowledgment for the UI frame carrying the Hello message to the joining station.

12. Station 25 forwards the Hello message throughout the network using Intranet Relay.

E.6.4.2.2 Message Formats.

1. Join Request Message to Network Controller

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	yyy	(station's current #)
T-bits	00	(No Info)
Link Layer Header (bits, LSB on right):		
Source Address	00000010	(Net Entry)
Destination Address	00000101	(network controller)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Message Number	20	(Join Request)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

2. Join Request Message to Global Multicast Address

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	000	(Initial)
T-bits	00	(No Info)
Link Layer Header (bits, LSB on right):		
Source Address	00000010	(Net Entry)
Destination Address	11111111	(Global Multicast)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	1	(Net Entry)
Forwarder Address	0	(Unknown)
Destination Address 2	(network controller)	
Message Number	20	(Join Request)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

3. Delay Time Message

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	yyy	(station's current #)
T-bits	01	(DAP-NAD)
Data Link Precedence	01	Priority)
First Subscriber Number	zzzzzz	(next ranked station)
Link Layer Header (bits, LSB on right):		
Source Address	0ffffff0	(forwarder's address)
Destination Address	00000011	(Net Entry)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	00ffffff	(Forwarder)
Forwarder Address	00ffffff	(Forwarder)
Destination Address	1	(Net Entry)
Message Number	27	(Delay Time)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Time	tttttttt	(Seconds)
Terminator Block	255	

4. Join Request Message to Forwarder

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	000	(Initial)
T-bits	00	(No Info)
Link Layer Header (bits, LSB on right):		
Source Address	00000010	(Net Entry)
Destination Address	00110011	(forwarder #25)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	1	(Net Entry)
Forwarder Address	25	(station #25)
Destination Address	2	(network controller)
Message Number	20	(Join Request)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

5. Join Request Message to Network Controller from Forwarder

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	yyy	(station's current #)
T-bits	01	(DAP-NAD)
Data Link Precedence	01	(Priority)
First Subscriber Number	000010	(network controller)
Link Layer Header (bits, LSB on right):		
Source Address	00110010	(forwarder #25)
Destination Address	00000101	(network controller)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	1	(Net Entry)
Forwarder Address	25	(station #25)
Destination Address	2	(network controller)
Message Number	20	(Join Request)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

6. Parameter Update Message

Transmission Header (bits, LSB on right):

Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	000	(Initial)
T-bits	01	(DAP-NAD)
Data Link Precedence	00	(Urgent)
First Subscriber Number	000010	(network controller)

Link Layer Header (bits, LSB on right):

Source Address	00000100	(network controller)
Destination Address(es)	[up to 16 data link addresses]	
Control Field	00010011	(UI, ACK required)

Intranet Header (octets):

Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	240	(11110000: Network Control, low delay)

XNP Message (octets):

Message Number	26	(Parameter Update)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
XNP Data Blocks		
Data Block 1	Station Identification (station #1)	
Data Block 11	NAD Ranking (station #1)	
•		
•		
•		
Data Block 1	Station Ident. (last station)	
Data Block 11	NAD Ranking (last station)	
Terminator Block	255	

7. Type 1 Acknowledgment to UI Carrying Parameter Update Message

Transmission Header (bits, LSB on right):

Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	yyy	(each station's current #)
T-bits	01	(DAP-NAD)
Data Link Precedence	00	(Urgent)
First Subscriber Number	zzzzzz	(using old ranking)

Link Layer Header (bits, LSB on right):

Source Address	0aaaaaa1	(Acknowledging station)
Destination Address	00000101	(network controller)
Control Field	00110011	(URR response)

8. Join Accept Message to Forwarder

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	000	(Initial)
T-bits	01	(DAP-NAD)
Data Link Precedence	01	(Priority)
First Subscriber Number	011001	(forwarder #25)
Link Layer Header (bits, LSB on right):		
Source Address	00000100	(network controller)
Destination Address	00110011	(forwarder #25)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	240	(Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	2	(network controller)
Forwarder Address	25	(station #25)
Destination Address	1	(Net Entry)
Message Number	21	(Join Accept)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Address Map	[one bit set to 0, specifying the joiner's link address	
Comm. Functions	1	(Centralized)
XNP Data Blocks:		
Data Block 2	Basic Network Parameters	
Data Block 4	Type 3 Network Parameters	
Data Block 5	Deterministic NAD Parameters	
Data Block 9	Type 2 Parameters	
Data Block 12	Intranet Parameters	
Data Block 1	Station Identification (station #1)	
Data Block 11	NAD Ranking (station #1)	
• • •		
Data Block 1	Station Ident. (last station)	
Data Block 11	NAD Ranking (last station)	
Terminator Block	255	

9. Join Accept Message to Joiner from Forwarder

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	yyy	(forwarder's current #)
T-bits	01	(DAP-NAD)
Data Link Precedence	01	(Priority)
First Subscriber Number	000001	(Net Entry)
Link Layer Header (bits, LSB on right):		
Source Address	00110010	(forwarder #25)
Destination Address	00000011	(Net Entry)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	2	(network controller)
Forwarder Address	25	(station #25)
Destination Address	1	(Net Entry)
Message Number	21	(Join Accept)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Address Map	[one bit set to 0, specifying the joiner's link address]	
Comm. Functions	1	(Centralized)
XNP Data Blocks:		
Data Block 2	Basic Network Parameters	
Data Block 4	Type 3 Network Parameters	
Data Block 5	Deterministic NAD Parameters	
Data Block 9	Type 2 Parameters	
Data Block 12	Intranet Parameters	
Data Block 1	Station Identification (station #1)	
Data Block 11	NAD Ranking (station #1)	
• • •		
Data Block 1	Station Ident. (last station)	
Data Block 11	NAD Ranking (last station)	
Terminator Block	255	

10. Hello Message from Joiner

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	000	(Initial)
T-bits	01	(DAP-NAD)
Data Link Precedence	00	(Urgent)
First Subscriber Number	011001	(forwarder #25)
Link Layer Header (bits, LSB on right):		
Source Address	0xxxxxx0	(Joiner)
Destination Addresses	00110010	(forwarder #25)
	11111111	(Global Multicast)
Control Field	00010011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	XX	(Joiner)
Forwarder Address	25	(station #25)
Destination Address	127	(Global Broadcast)
Message Number	23	(Hello)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

11. Type 1 Acknowledgment to UI Carrying Hello Message

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	yyy	(each station's current #)
T-bits	01	(DAP-NAD)
Data Link Precedence	00	(Urgent)
First Subscriber Number	zzzzzz	(using old ranking)
Link Layer Header (bits, LSB on right):		
Source Address	00110010	(forwarder #25))
Destination Address	0xxxxxx1	(Joiner)
Control Field	00110011	(URR response)

12. Hello Message from Forwarder

Transmission Header (bits, LSB on right):		
Selection Bits	011	(FEC/TDC/No Scrambling)
Topology Update ID	000	(Initial)
T-bits	01	(DAP-NAD)
Data Link Precedence	00	(Urgent)
First Subscriber Number	011001	(forwarder #25)
Link Layer Header (bits, LSB on right):		
Source Address	00110010	(forwarder #25)
Destination Addresses	11111111	(Global Multicast)
Control Field	00100011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum, assuming Relaying is not required)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	XX	(Joiner)
Forwarder Address	25	(station #25)
Destination Address	127	(Global Broadcast)
Message Number	23	(Hello)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

E.6.4.3 Distributed Network Control, All Stations Are Network Controllers. In this example, the network is using multiple distributed network controllers, and all stations in the network have network controller capabilities. The network is using data link Types 1, 3 and 4, and RE-NAD. The joining station has all optional capabilities. Therefore the sequence of events is shown in Figure E-6 and is described in section E.6.4.3.1. Detailed message formats are provided in section E.6.4.3.2.

E.6.4.3.1 Sequence of Events.

1. The joining station sends a UI frame with a Join Request message to the network controller requesting entry to the network. No data blocks are appended since the joining station does not have knowledge of hardware parameters, but does have all optional capabilities.

2. One or more network controllers respond with a Join Accept message to the joiner. The Communications Functions field in the Join Accept message is set to Distributed, and the Address Map has bits set to 0 for all available addresses. Data block 2, block 4, block 7, block 10 and block 12 are appended to the Join Accept message to provide the basic operating parameters to joining station. Block 8 is appended to specify a Wait Timer that the joiner must use to verify that its selected data link address is accepted.

3. The joining station selects an address from the Address Map provided in the Join Accept message and sends a Hello message to validate the selected address. Since the joining station has no knowledge of topology, the Hello message is sent with a forwarding header to one of the network controllers which responded with a Join Accept message. The UI frame carrying the Hello message has the P-bit set to one, uses the selected data link address as the source address, and uses the selected network controller address and the Global multicast address as destinations.

4. The selected network controller sends a Type 1 Acknowledgment in response to the UI frame carrying the Hello message. The joining station starts the Wait Timer.

5. The selected network controller forwards the Hello message to all other stations.

6. If another station determines that the selected address is already in use, it sends a Join Reject message through the forwarding network controller to the joiner.

7. The forwarding network controller sends the Join Reject message to the joining station and broadcasts the Join Reject to all network members.

8. The joining station selects a new address from the address map and sends another Hello message to validate the selected address. This address is also sent through the forwarding network controller.

9. The selected network controller sends another Type 1 Acknowledgment in response to the UI frame carrying the Hello message. The joining station restarts the Wait Timer.

10. The selected network controller again forwards the Hello message to all other stations.

If the address selected by the joiner is unique, no Join Reject will be provided. The joiner has successfully entered the network after the Wait Timer expires.

E.6.4.3.2 Message Formats.

1. Join Request Message

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	000	(Initial)
T-bits	00	(No Info)
Link Layer Header (bits, LSB on right):		
Source Address	00000010	(Net Entry)
Destination Address	00000101	(network controller)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Message Number	20	(Join Request)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

2. Join Accept Message

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	000	(Initial)
T-bits	10	(RE-NAD)
Queue Precedence	01	(Priority)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)
Link Layer Header (bits, LSB on right):		
Source Address	nnnnnnn0	(network controller's link address)
Destination Address	00000011	(Net Entry)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Message Number	21	(Join Accept)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Address Map	[bits set to 0, specifying available data link addresses]	
Comm. Functions	2	(Distributed)
XNP Data Blocks:		
Data Block 2	Basic Network Parameters	
Data Block 4	Type 3 Network Parameters	
Data Block 7	RE-NAD Parameters	
Data Block 10	Type 4 Parameters	
Data Block 12	Intranet Parameters	
Data Block 8	Wait Time	
Terminator Block	255	

3. Hello Message from Joiner

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	000	(Initial)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	0000	
Link Layer Header (bits, LSB on right):		
Source Address	0xxxxxx0	(Joiner)
Destination Addresses	yyyyyyy0	(selected network controller)
	11111111	(Global Multicast)
Control Field	00010011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	XX	(Joiner's selected address)
Forwarder Address	YY	(selected network controller)
Destination Address	127	(Global Broadcast)
Message Number	23	(Hello)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

4. Type 1 Acknowledgment to UI Carrying Hello Message

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	000	(Initial)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	0000	
Link Layer Header (bits, LSB on right):		
Source Address	yyyyyyy0	(Forwarding network controller)
Destination Address	0xxxxxx1	(Joiner)
Control Field	00110011	(URR response)

5. Hello Message from Forwarding Network Controller

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	zzz	(network controller's current #)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	0000	
Link Layer Header (bits, LSB on right):		
Source Address	yyyyyyy0	(selected network controller)
Destination Addresses	11111111	(Global Multicast)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum, if no Relay required)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	XX	(Joiner)
Forwarder Address	YY	(selected network controller)
Destination Address	127	(Global Broadcast)
Message Number	23	(Hello)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

6. Join Reject Message to Forwarding Network Controller

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	zzz	(station's current #)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	0000	
Link Layer Header (bits, LSB on right):		
Source Address	00000100	(network controller)
Destination Address	yyyyyyy1	(Forwarding network controller)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum, if no Relay required)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	2	(network controller)
Forwarder Address	Y	(forwarder's address)
Destination Address	X	(address selected by joiner)
Message Number	22	(Join Reject)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Rejected Link Address	X	(address selected by joiner)
Address Map	[bits set to 0, specifying available data link addresses]	
Comm. Functions	2	(Distributed)
Terminator Block	255	

7. Join Reject Message to Joiner from Forwarding Network Controller

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	zzz	(forwarder's current #)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	0000	
Link Layer Header (bits, LSB on right):		
Source Address	yyyyyyy0	(forwarder)
Destination Address	0xxxxxx0	(Rejected Joiner)
	11111111	(Global Multicast)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	2	(network controller)
Forwarder Address	Y	(forwarder's address)
Destination Address	X	(address selected by joiner)
Message Number	22	(Join Reject)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Rejected Link Address	X	(address selected by joiner)
Address Map	[bits set to 0, specifying available data link addresses]	
Comm. Functions	2	(Distributed)
Terminator Block	255	

8. Second Hello Message from Joiner

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	000	(Initial)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	0000	
Link Layer Header (bits, LSB on right):		
Source Address	0xxxxxx0	(Joiner)
Destination Addresses	yyyyyyy0	(selected network controller)
	11111111	(Global Multicast)
Control Field	00010011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	XX	(Joiner)
Forwarder Address	YY	(selected network controller)
Destination Address	127	(Global Broadcast)
Message Number	23	(Hello)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

9. Type 1 Acknowledgment to UI Carrying Second Hello Message

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	000	(Initial)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	0000	
Link Layer Header (bits, LSB on right):		
Source Address	yyyyyyy0	(Forwarding network controller)
Destination Address	0xxxxxx1	(Joiner)
Control Field	11001100	(URR response)

10. Second Hello Message from Forwarding Network Controller

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	zzz	(forwarder's current #)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	0000	
Link Layer Header (bits, LSB on right):		
Source Address	yyyyyyy0	(forwarder)
Destination Addresses	11111111	(Global Multicast)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum, if no Relay required)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	XX	(Joiner)
Forwarder Address	YY	(selected network controller)
Destination Address	127	(Global Broadcast)
Message Number	23	(Hello)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

E.6.4.4 Distributed Network Control, Disconnected Joiner. In this example, the network is using multiple distributed network controllers, although all stations are not network controllers. The joiner in this example is not in direct line of sight to a network controller. The network is using data link Types 1, 3 and 4, and RE-NAD. The joining station has all optional capabilities. Therefore the sequence of events is shown in Figure E-9 and is described in section E.6.4.4.1. Detailed message formats are provided in section E.6.4.4.2.

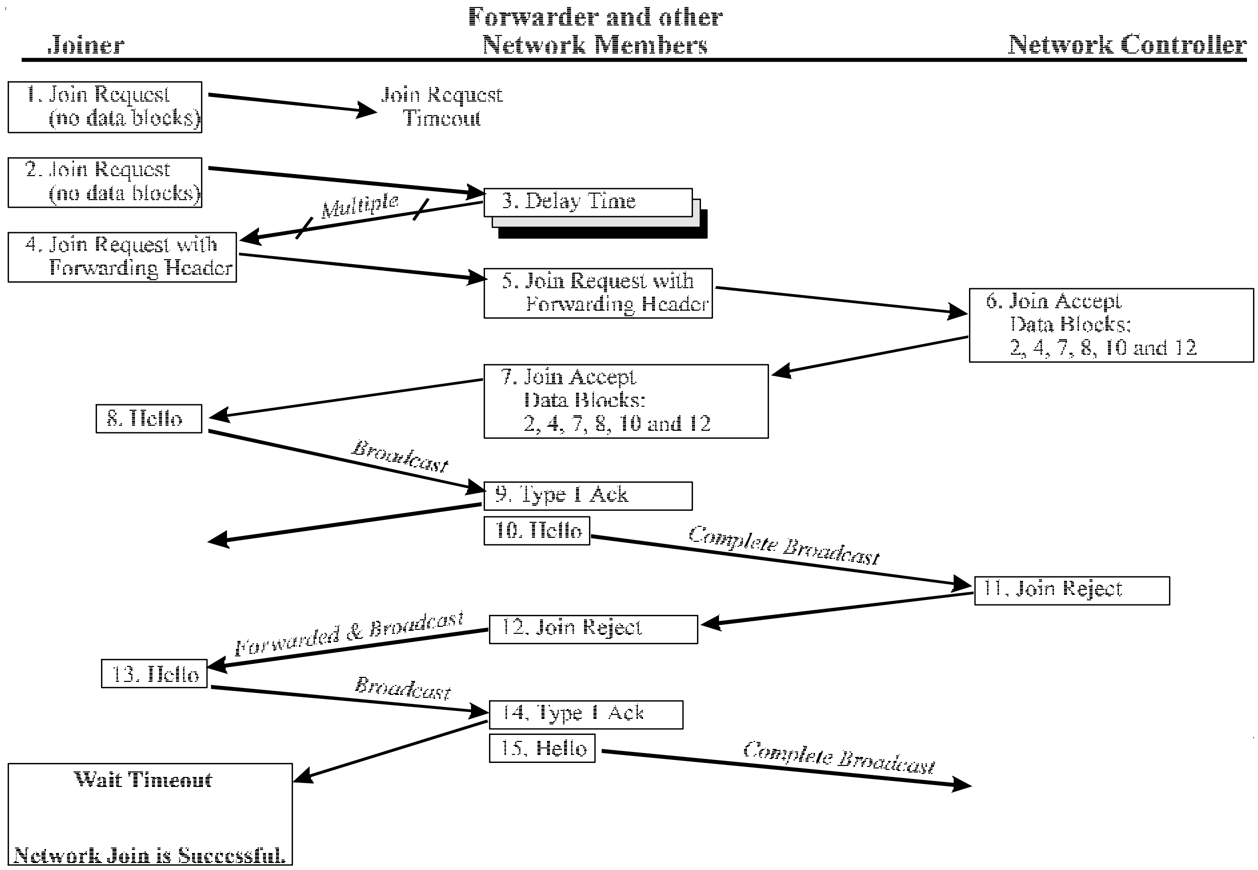


FIGURE E-9. Joining a disconnected, distributed network.

E.6.4.4.1 Sequence of Events.

1. The joining station sends a UI frame with a Join Request message to the network controller requesting entry to the network. No data blocks are appended since the joining station does not have knowledge of hardware parameters, but does have all optional capabilities.

   Because the joiner is not in direct line of sight with the network controller, network controller does not receive the Join Request and there is no response.

2. The joining station sends a UI Command with a Join Request message to the Global Multicast data link address requesting entry to the network. This Join Request message has a Forwarding Header identifying the network controller as the Destination.

3. The Join Request message is received by stations 44, 25 and 31. These three stations have a path to a network controller and send a Delay Time message to the joining station.

4. The joining station selects station 25 as the forwarder and uses this station to forward a Join Request message to the network controller.

5. Station 25 forwards the Join Request message to a network controller for the joining station. Station 25 will use the network controller's individual address at the data link layer and employ appropriate relaying to reach the network controller at the Intranet layer.

6. The network controller responds with a Join Accept message to the joiner, using station 25 as a forwarder. The Communications Functions field in the Join Accept message is set to Distributed, and the Address Map has bits set to 0 for all available addresses. Data block 2, block 4, block 7, block 10 and block 12 are appended to the Join Accept message to provide the basic operating parameters to the joining station. Block 8 is appended to specify a Wait Timer that the joiner must use to verify that its selected data link address is accepted.

7. Station 25 forwards network controller's Join Accept message (and all data blocks) to the joining station.

8. The joining station selects an address from the Address Map provided in the Join Accept message and sends a Hello message to validate the selected address. This Hello message is broadcast locally, and also is addressed to forwarding station 25 so that it can be broadcast completely through the network. The UI frame carrying the Hello message has the P-bit set to one, uses the selected data link address as the source address, and uses the selected forwarder's address and the Global multicast address as destinations.

9. Forwarding station 25 sends a Type 1 Acknowledgment for the UI frame carrying the Hello message to the joining station. The joining station starts the Wait Timer.

10. Station 25 forwards the Hello message throughout the network using Intranet Relay.

11. If another station determines that the selected address is already in use, it sends a Join Reject message through the forwarder to the joiner.

12. The forwarder sends the Join Reject message to the joining station and broadcasts the Join Reject to all network members.

13. The joining station selects a new address from the address map and sends another Hello message to validate the selected address. This address is also sent through the forwarder, station 25.

14. Forwarding station 25 sends another Type 1 Acknowledgment in response to the UI frame carrying the Hello message. The joining station restarts the Wait Timer.

15. Station 25 again forwards the Hello message to all other stations using Intranet Relay.

If the address selected by the joiner is unique, no Join Reject will be provided. The joiner has successfully entered the network after the Wait Timer expires.

E.6.4.4.2 Message Formats.

1. Join Request Message

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	000	(Initial)
T-bits	00	(No Info)
Link Layer Header (bits, LSB on right):		
Source Address	00000010	(Net Entry)
Destination Address	00000101	(network controller)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Message Number	20	(Join Request)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

2. Join Request Message to Global Multicast address

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	000	(Initial)
T-bits	00	(No Info)
Link Layer Header (bits, LSB on right):		
Source Address	00000010	(Net Entry)
Destination Address	11111111	(Global Multicast)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	1	(Net Entry)
Forwarder Address	0	(Unknown)
Destination Address	2	(network controller)
Message Number	20	(Join Request)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

3. Delay Time Message

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	yyy	(station's current #)
T-bits	10	(RE-NAD)
Queue Precedence	01	(Priority)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)
Link Layer Header (bits, LSB on right):		
Source Address	0ffffff0	(forwarder's address)
Destination Address	00000011	(Net Entry)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	00ffffff	(Forwarder)
Forwarder Address	00ffffff	(Forwarder)
Destination Address	1	(Net Entry)
Message Number	27	(Delay Time)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Time	tttttttt	(Seconds)
Terminator Block	255	

4. Join Request Message to Forwarder

Transmission Header (bits, LSB on right):		
Selection Bits	000	(FEC/TDC/No Scrambling)
Topology Update ID	000	(Initial)
T-bits	00	(No Info)
Link Layer Header (bits, LSB on right):		
Source Address	00000010	(Net Entry)
Destination Address	00110011	(forwarder #25)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	1	(Net Entry)
Forwarder Address	25	(station #25)
Destination Address	2	(network controller)
Message Number	20	(Join Request)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Station Identifier	[24-bit Unit Reference Number, followed by 8 zero-bits]	
Terminator Block	255	

5. Join Request Message to Network Controller from Forwarder

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	yyy	(station's current #)
T-bits	10	(RE-NAD)
Queue Precedence	01	(Priority)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)
Link Layer Header (bits, LSB on right):		
Source Address	00110010	(forwarder #25)
Destination Address	00000101	(network controller)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	1	(Net Entry)
Forwarder Address	25	(station #25)
Destination Address	2	(network controller)
Message Number	20	(Join Request)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

6. Join Accept Message to Forwarder

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	yyy	(station's current #)
T-bits	10	(RE-NAD)
Queue Precedence	01	(Priority)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)
Link Layer Header (bits, LSB on right):		
Source Address	nnnnnnn0	(network controller's link address)
Destination Address	00110011	(forwarder #25)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	2	(network controller)
Forwarder Address	25	(station #25)
Destination Address	1	(Net Entry)
Message Number	21	(Join Accept)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Address Map	[bits set to 0, specifying available data link addresses]	
Comm. Functions	2	(Distributed)
XNP Data Blocks:		
Data Block 2	Basic Network Parameters	
Data Block 4	Type 3 Network Parameters	
Data Block 7	RE-NAD Parameters	
Data Block 10	Type 4 Parameters	
Data Block 12	Intranet Parameters	
Data Block 8	Wait Time	
Terminator Block	255	

7. Join Accept Message to Joiner from Forwarder

Transmission Header (bits, LSB on right):

Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	yyy	(station's current #)
T-bits	10	(RE-NAD)
Queue Precedence	01	(Priority)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)

Link Layer Header (bits, LSB on right):

Source Address	00110010	(forwarder #25)
Destination Address	00000011	(Net Entry)
Control Field	00000011	(UI)

Intranet Header (octets):

Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	48	(00110000: Priority, low delay)

XNP Message (octets):

Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	2	(network controller)
Forwarder Address	25	(station #25)
Destination Address	1	(Net Entry)
Message Number	21	(Join Accept)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Address Map	[bits set to 0, specifying available data link addresses]	
Comm. Functions	2	(Distributed)

XNP Data Blocks:

Data Block 2	Basic Network Parameters
Data Block 4	Type 3 Network Parameters
Data Block 7	RE-NAD Parameters
Data Block 10	Type 4 Parameters
Data Block 12	Intranet Parameters
Data Block 8	Wait Time
Terminator Block	255

8. Hello Message from Joiner

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	000	(station's current #)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)
Link Layer Header (bits, LSB on right):		
Source Address	0xxxxxx0	(Joiner)
Destination Addresses	00110010	(forwarder #25)
	11111111	(Global Multicast)
Control Field	00010011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	240	(Network Control. low delay)
XNP Message (octets):		
Version Number	0	(current version)
Forwarding Header:		
Message Number	0	(Forwarding Header)
Source Address	XX	(Joiner)
Forwarder Address	25	(station #25)
Destination Address	127	(Global Broadcast)
Message Number	23	(Hello)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

9. Type 1 Acknowledgment to UI Carrying Hello Message

Transmission Header (bits, LSB on right):

Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	yyy	(station's current #)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)

Link Layer Header (bits, LSB on right):

Source Address	00110010	(forwarder #25))
Destination Address	0xxxxxx1	(Joiner)
Control Field	00110011	(URR response)

10. Hello Message from Forwarder

Transmission Header (bits, LSB on right):

Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	yyy	(station's current #)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)

Link Layer Header (bits, LSB on right):

Source Address	00110010	(forwarder #25)
Destination Addresses	11111111	(Global Multicast)
Control Field	00010011	(UI)

Intranet Header (octets):

Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum, assuming Relaying is not required)
Type of Service	240	(11110000: Network Control, low delay)

XNP Message (octets):

Version Number	0	(current version)
Message Number	0	(Forwarding Header)
Source Address	XX	(Joiner)
Forwarder Address	25	(station #25)
Destination Address	127	(Global Broadcast)
Message Number	23	(Hello)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

11. Join Reject Message to Forwarder

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	zzz	(station's current #)
T-bits	01	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)
Link Layer Header (bits, LSB on right):		
Source Address	00000100	(network controller)
Destination Address	00110011	(forwarder #25)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum, if no Relay required)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Message Number	0	(Forwarding Header)
Source Address	2	(network controller)
Forwarder Address	25	(station 25)
Destination Address	X	(address selected by joiner)
Message Number	22	(Join Reject)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Rejected Link Address	X	(address selected by joiner)
Address Map	[bits set to 0, specifying available data link addresses]	
Comm. Functions	2	(Distributed)
Terminator Block	255	

12. Join Reject Message to Joiner from Forwarder

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	zzz	(network controller's current #)
T-bits	01	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)
Link Layer Header (bits, LSB on right):		
Source Address	00110010	(forwarder #25)
Destination Address	0xxxxxx0	(Rejected Joiner)
	11111111	(Global Multicast)
Control Field	00000011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Message Number	0	(Forwarding Header)
Source Address	2	(network controller)
Forwarder Address	25	(station 25))
Destination Address	X	(address selected by joiner)
Message Number	22	(Join Reject)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Rejected Link Address	X	(address selected by joiner)
Address Map	[bits set to 0, specifying available data link addresses]	
Comm. Functions	2	(Distributed)
Terminator Block	255	

13. Second Hello Message from Joiner

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	000	(station's current #)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)
Link Layer Header (bits, LSB on right):		
Source Address	0xxxxxx0	(Joiner)
Destination Addresses	00110010	(forwarder #25)
	11111111	(Global Multicast)
Control Field	00010011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Message Number	0	(Forwarding Header)
Source Address	XX	(Joiner)
Forwarder Address	25	(station #25)
Destination Address	127	(Global Broadcast)
Message Number	23	(Hello)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

14. Type 1 Acknowledgment to UI Carrying Second Hello Message

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	yyy	(station's current #)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)
Link Layer Header (bits, LSB on right):		
Source Address	00110010	(forwarder #25))
Destination Address	0xxxxxx1	(Joiner)
Control Field	00110011	(URR response)

15. Second Hello Message from Forwarder

Transmission Header (bits, LSB on right):		
Selection Bits	000	(No FEC, TDC or Scrambling)
Topology Update ID	yyy	(station's current #)
T-bits	10	(RE-NAD)
Queue Precedence	00	(Urgent)
Queue Length	00zzzz	(network accesses needed to empty the transmit queue)
Link Layer Header (bits, LSB on right):		
Source Address	00110010	(forwarder #25)
Destination Addresses	11111111	(Global Multicast)
Control Field	00010011	(UI)
Intranet Header (octets):		
Version #	0	(Current Version)
Message Type	6	(XNP)
Header Length	3	(Minimum, assuming Relaying is not required)
Type of Service	240	(11110000: Network Control, low delay)
XNP Message (octets):		
Version Number	0	(current version)
Message Number	0	(Forwarding Header)
Source Address	XX	(Joiner)
Forwarder Address	25	(station #25)
Destination Address	127	(Global Broadcast)
Message Number	23	(Hello)
Station Identifier	[in the least significant 24-bits, Unit Reference Number of joiner]	
Terminator Block	255	

APPENDIX F

GOLAY CODING ALGORITHM

F.1. General.

F.1.1 Scope. This appendix contains amplifying information in support of MIL-STD-188-220.

F.1.2 Application. This appendix is not a mandatory part of MIL-STD-188-220. The information contained herein is intended for guidance only.

F.2. Applicable documents. None.

F.3. Forward error correction. The FEC method requires the receiver to detect and automatically correct errors in a received block of information. The number of errors the receiver can detect and correct depends on the coding method. The information bits (k) are separated into blocks that contain both information bits and code bits. The length of the block, including the information and code bits, is (n). The code is described as (n,k), where n is the length of the block and k is the number of information bits in the block.

F.4. Golay code. The Golay code is a linear, block, perfect, and cyclic (23,12) code capable of correcting any combination of three or fewer errors in a block of 23 digits. The generator polynomial for this code is

$$g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

where g(x) is a factor of $x^{23} + 1$

F.4.1 Half-rate Golay code. The half-rate Golay code (24,12) is formed by adding a fill bit to the Golay (23, 12) code. The fill bit is not checked on reception. The (24,12) code is preferable to the (23, 12) because it has a code rate of exactly one-half. This code rate simplifies system timing.

F.4.2 Golay code implementation. The Golay code may be implemented in either hardware or software. The hardware implementation uses shift-registers for encoding and decoding, as described in F.4.2.1 and F.4.2.2, respectively. The software implementation uses a generator matrix and conversion table, as described in F.4.2.3.

F.4.2.1 Hardware implementation. Golay code encoding can be performed with an 11-stage feedback shift register with feedback connections selected according to the coefficients of g(x). A shift register corresponding to the coefficients of g(x) is shown in Figure F-1. The k information bits are located at the beginning of the n symbol block code. With the gate open, the information bits are loaded into the shift register stages and simultaneously into the output channel. At this time the shift register contains the check symbols. With the gate closed, register contents are then shifted onto the output channel. The last n - k symbols are the check symbols that form the whole codeword.

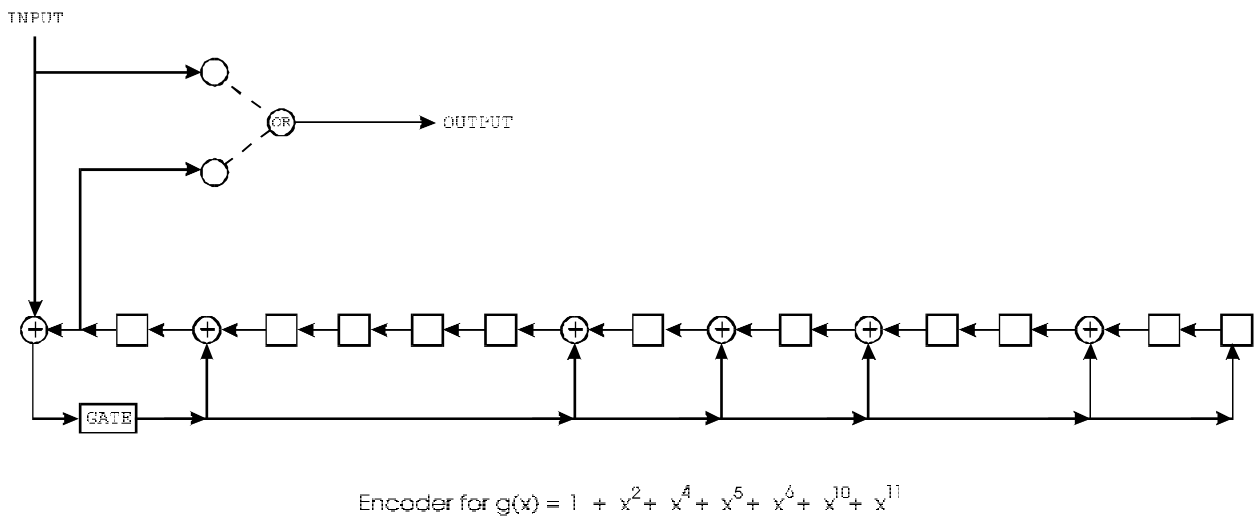


FIGURE F-1. Shift register encoding for the (23, 12) Golay code.

F.4.2.2 Hardware decoding. The Golay code is decoded using a number of techniques such as the error-trapping process developed by T. Kasami. The Kasami error-trapping decoder for the Golay code is shown in Figure F-2. It works as follows:

a. Gates 1, 3, and 5 are opened, and gates 2 and 4 are closed. The received codeword r(x) is then shifted into both the 23-stage shift register and the syndrome register. At the same time, the previously corrected codeword is shifted out to the user. The syndrome

$$S(x) = S_0 + S_1x + \ldots + S_{10}x^{10}$$

is then formed and subjected to threshold tests.

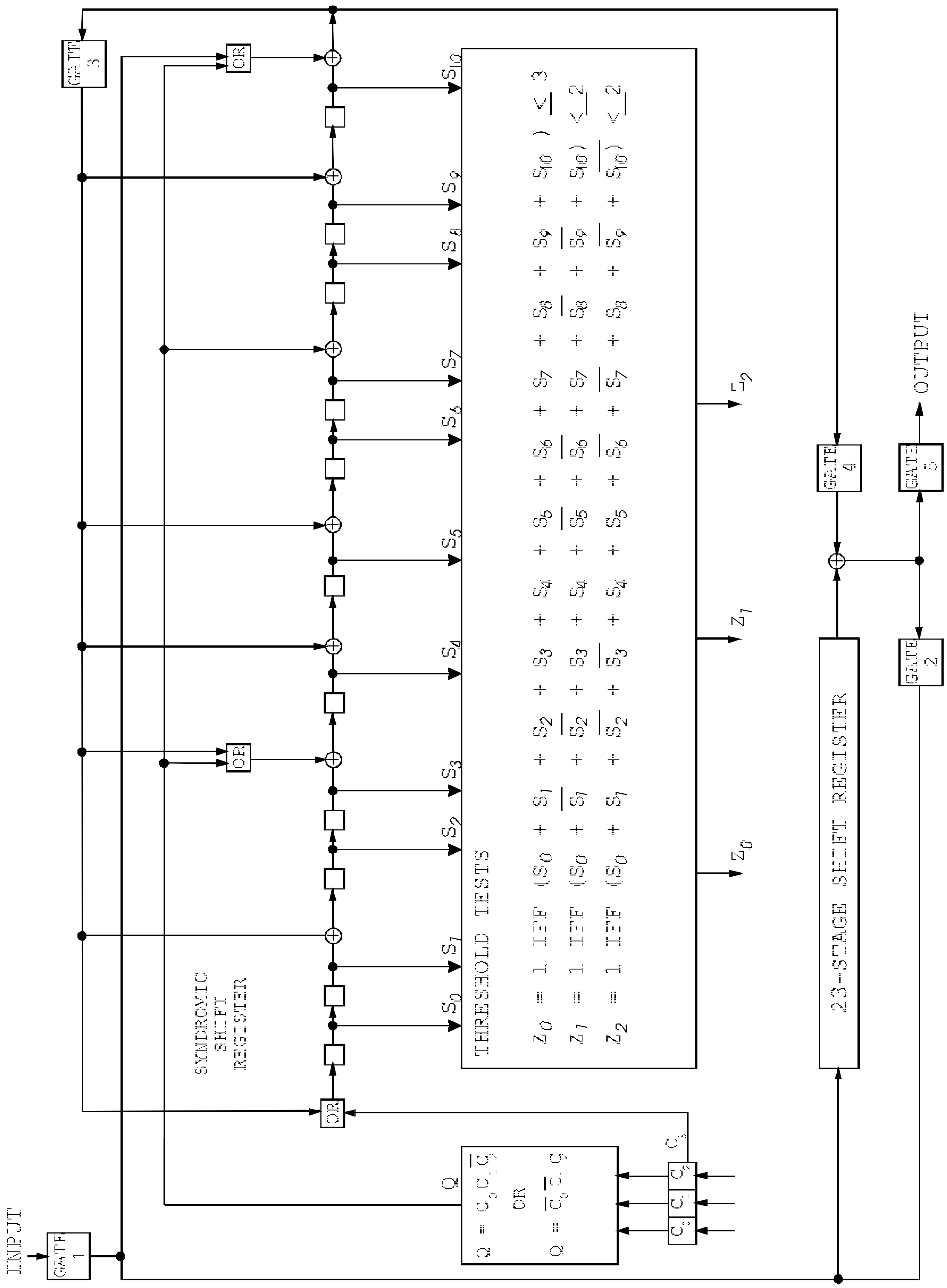


FIGURE F-2. <u>Kasami error-trapping decoder for the (24, 12) Golay code</u>.

b. Gates 1, 4, and 5 are closed and gate 2 is opened. Gate 3 remains open. The threshold tests occur in the following order:

1. If Z_0 is unity, then all the errors are confined to the 11 high-order positions of r(x), and S(x) matches the errors. Z_0 opens gate 4 and closes gate 3. Contents of both the 23-stage shift register and the syndrome shift register are then shifted 11 times, and the errors are corrected. Then gate 4 is closed and the contents of the 23-stage shift register are shifted until the received codeword is in its original position. The decoder then goes to step 3 below.

2. If Z_1 is unity, the error pattern in S(x) is the same as the errors in the 11 high-order bits of the codeword r(x), and a single error exists at location x^5. Gate 4 is opened and gate 3 is closed. The counter is preloaded with a count of 2, and both the syndrome shift register and the 23-stage shift register are shifted until the error in x^5 is corrected. Then gate 4 is closed, and the contents of the 23-stage shift register are shifted until the received codeword is in its original position. The decoder then goes to step 3.

3. If Z_2 is unity, the error pattern in S(x) is the same as the errors in the 11 high-order bits of the codeword r(x), and there is a single error in location x^6. The same steps are followed as in b (above) except that the counter is preloaded with a count of 3. The decoder then goes to step 3.

4. If neither of the three thresholds is unity, the decoder goes directly to step 3.

c. Gates 1, 4, and 5 are closed, and gates 2 and 3 are opened. Contents of both the 23-stage shift register and the syndrome shift register are then shifted once to the right. The decoder then goes to step 2.

4d This action continues until step 3 has been executed 46 times. Then the decoder returns to step 1 to process the next received codeword.

The decoder always yields an output. The output is correct if there were 3 or fewer errors in the received codeword, and erroneous if there were more than 3 errors in the codeword.

F.4.2.3 Software implementation. The transmitting DMTD shall generate the check bits using the following generator polynomial:

$$g(x)= x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

Note that using modulo 2 addition,

$$x^{23}+1=(x^{11}+x^{10}+x^{6}+x^{5}+x^{4}+x^{2}+1)(x^{11}+x^{9}+x^{7}+x^{6}+x^{5}+x+1)(x+1)$$

The 11 check bits shall be as derived from the generator matrix G, shown in Figure F-3, where the matrix contains the coefficients of the polynomials on the left.

$$\begin{array}{r@{\;}l}
 & \begin{matrix} x^{22} & x^{21} & x^{20} & x^{19} & x^{18} & x^{17} & x^{16} & x^{15} & x^{14} & x^{13} & x^{12} & x^{11} & x^{10} & x^{9} & x^{8} & x^{7} & x^{6} & x^{5} & x^{4} & x^{3} & x^{2} & x^{1} & x^{0} \end{matrix} \\
\begin{array}{r}
x^{11} * g(x) = \\
x^{10} * g(x) = \\
x^{9} * g(x) + x^{11} * g(x) = \\
x^{8} * g(x) + x^{10} * g(x) = \\
x^{7} * g(x) + x^{9} * g(x) = \\
(x^{6} + x^{8} + x^{11}) * g(x) = \\
(x^{5} + x^{7} + x^{10}) * g(x) = \\
(x^{4} + x^{6} + x^{9}) * g(x) = \\
(x^{3} + x^{5} + x^{8} + x^{11}) * g(x) = \\
(x^{2} + x^{4} + x^{7} + x^{10} + x^{11}) * g(x) = \\
(x + x^{3} + x^{6} + x^{9} + x^{10} + x^{11}) * g(x) = \\
(1 + x^{2} + x^{5} + x^{8} + x^{9} + x^{10} + x^{11}) * g(x) =
\end{array} &
\underbrace{\left[\begin{matrix}
1&1&0&0&0&1&1&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0 \\
0&1&1&0&0&0&1&1&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0 \\
1&1&1&1&0&1&1&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\
0&1&1&1&1&0&1&1&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0 \\
0&0&1&1&1&1&0&1&1&0&1&0&0&0&0&1&0&0&0&0&0&0&0 \\
1&1&0&1&1&0&0&1&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\
0&1&1&0&1&1&0&0&1&1&0&0&0&0&0&0&0&1&0&0&0&0&0 \\
0&0&1&1&0&1&1&0&0&1&1&0&0&0&0&0&0&0&1&0&0&0&0 \\
1&1&0&1&1&1&0&0&0&1&1&0&0&0&0&0&0&0&0&1&0&0&0 \\
1&0&1&0&1&0&0&1&0&1&1&0&0&0&0&0&0&0&0&0&1&0&0 \\
1&0&0&1&0&0&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&1&0 \\
1&0&0&0&1&1&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&1
\end{matrix}\right]}_{\text{Parity} \qquad\qquad\qquad \text{Identity}} = G
\end{array}$$

FIGURE F-3. Generator matrix G.

By interchanging the I and P columns to obtain matrix T, shown in Figure F-4, that is,

$$G=[P,I]_{(12\ x\ 23)} = >[I,P]_{(12\ x\ 23)} = T$$

the transmission order and value of the code word bits can be obtained by matrix multiplication (modulo 2 addition without carry) as follows:

$$\underset{(1x12)}{\left[Ib_1 \ \text{INFO BITS} \ Ib_{12} \right]} \cdot \underset{(12x23)}{\left[I,P \right]} = \underset{(1x23)}{\left[\text{INFO BITS,CHECK BITS} \right]}$$

FIRST BIT TRANSMITTED　　　　FIRST BIT TRANSMITTED

$$\mathbf{T} = \left[\underbrace{\begin{matrix}
1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&1
\end{matrix}}_{\mathbf{I}}\;\underbrace{\begin{matrix}
1&1&0&0&0&1&1&1&0&1&0\\
0&1&1&0&0&0&1&1&1&0&1\\
1&1&1&1&0&1&1&0&1&0&0\\
0&1&1&1&1&0&1&1&0&1&0\\
0&0&1&1&1&1&0&1&1&0&1\\
1&1&0&1&1&0&0&1&1&0&0\\
0&1&1&0&1&1&0&0&1&1&0\\
0&0&1&1&0&1&1&0&0&1&1\\
1&1&0&1&1&1&0&0&0&1&1\\
1&0&1&0&1&0&0&1&0&1&1\\
1&0&0&1&0&0&1&1&1&1&1\\
1&0&0&0&1&1&1&0&1&0&1
\end{matrix}}_{\mathbf{P}}\right]$$

FIGURE F-4. Matrix T.

APPENDIX G

PACKET CONSTRUCTION AND BIT ORDERING

G.1. General.

G.1.1 Scope. This appendix illustrates the construction of packets starting with the Application Layer Protocol Data Unit (PDU) and VMF Message data buffers and ending with the data link bit order of transmission and physical layer PDU. However this example excludes the S/R protocol. The focus of this example is to show correct formatting of the 188-220 subnetwork.

G.1.2 Application. This appendix is a mandatory part of this document. The bit ordering defined herein shall be utilized by all implementers.

G.2. Applicable Documents.

a. RFC 768: User Datagram Protocol

b. RFC 791: Internet Protocol -- DARPA Internet Program Protocol Specification

c. MIL-STD-2045-47001: Interoperability Standard for Connectionless Data Transfer -- Application Standard

d. Joint Interoperability of Tactical Command and Control Systems, Variable Message Format Technical Interface Design Plan (Test Edition), Reissue 2, Volume III

G.3. PDU Construction. This section provides examples illustrating the construction and bit ordering of a VMF message through the Application Layer, the Transport Layer, the Network Layer, Link Layer and Physical Layer. For clarity, each layer will be discussed separately and then combined for actual transmission. The same representations will be utilized for each layer:

- the MSB (2^n bit) is represented with an italicized font and
- the LSB (2^0 bit) is shown to the RIGHT in the Value (binary) column.

This representation is carried into the other columns to identify the beginning and end of each of the fields as the bits are moved into individual octets. Note that the bit markings for MSB and LSB are on a field basis, not on an octet basis. Single bit fields are treated as LSB. In addition, since some layers (e.g. transport) are based on commercial standards, the representation from the appropriate RFC will also be included. In all cases, we will start with a figure which illustrates the interaction with upper/lower communication layers, followed by a figure showing the exchange between communication layers. There will be a table showing the construction of the PDU. This will be followed by a table showing the construction of each octet and a figure showing the serial representation of this particular PDU as it would appear at physical layer.

Each layer typically adds value and its own header to an outgoing message. This process is illustrated in Figure G-1.

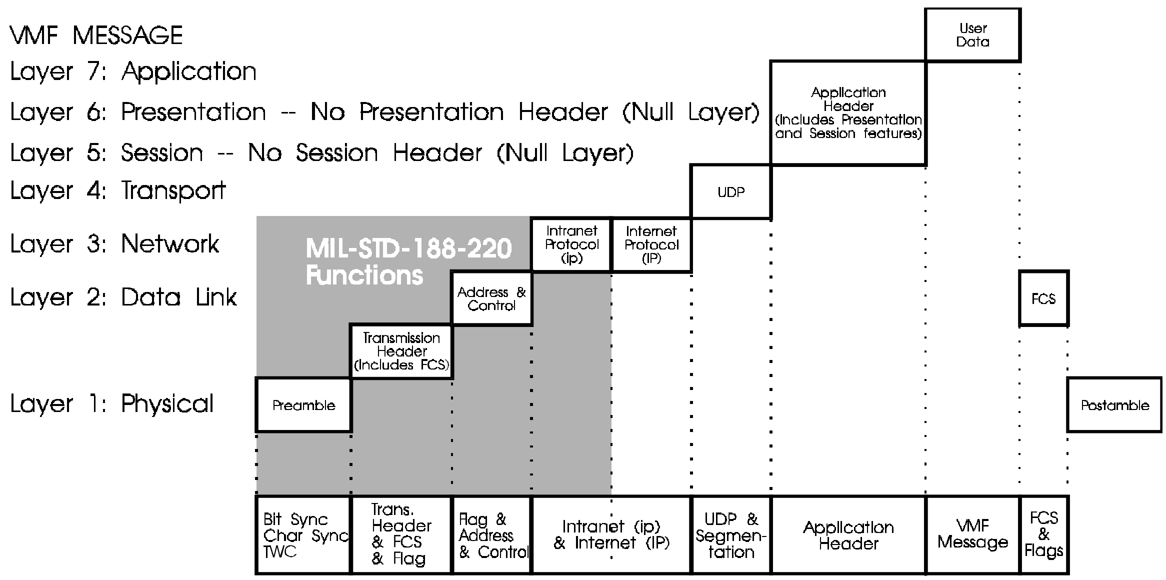


FIGURE G-1. PDU construction.

An application header is added to the VMF message at application layer. For this protocol, layers 5 and 6 are null layers, and no processing or headers are present. The Application Layer handles these functions. The transport layer adds its header. Although the standard calls out TCP, UDP and segmentation/reassembly, only UDP is illustrated in this appendix. Next, the network layer adds the IP header and the Intranet header. The message is now passed to the data link layer which adds both a header and a trailer. Finally, the physical layer adds its header resulting in the final PDU for transmission. Note that this example does not include TCP, segmentation & reassembly, or COMSEC.

G.3.1 VMF Message Data Exchange. The relationship of the VMF Messaging Services to other communication layers is shown in Figure G-2.

A layered communication model is used in this example for consistency with the principles of the ISO OSI reference model. The model discussed here is tailored to focus attention specifically on VMF Messaging Services, and the data it produces. A user of VMF Messaging Services exchanges Message Content with its peer at another node by sending and receiving the Message Content via the VMF Messaging Services. VMF Messaging Services sends and receives the Message Content by converting the Message Content to Message Data and exchanging the Message Data with its peer at another node. The VMF Message Data is sent and received via lower communication layers. The lower communication layers send and receive the VMF Message Data transparently over a variety of communications media. Note that VMF Messaging

Services would ordinarily use Application Layer services from the lower communication layers to send and receive Message Data. The Message Data would then appear in the Application Layer PDU's VMF message.

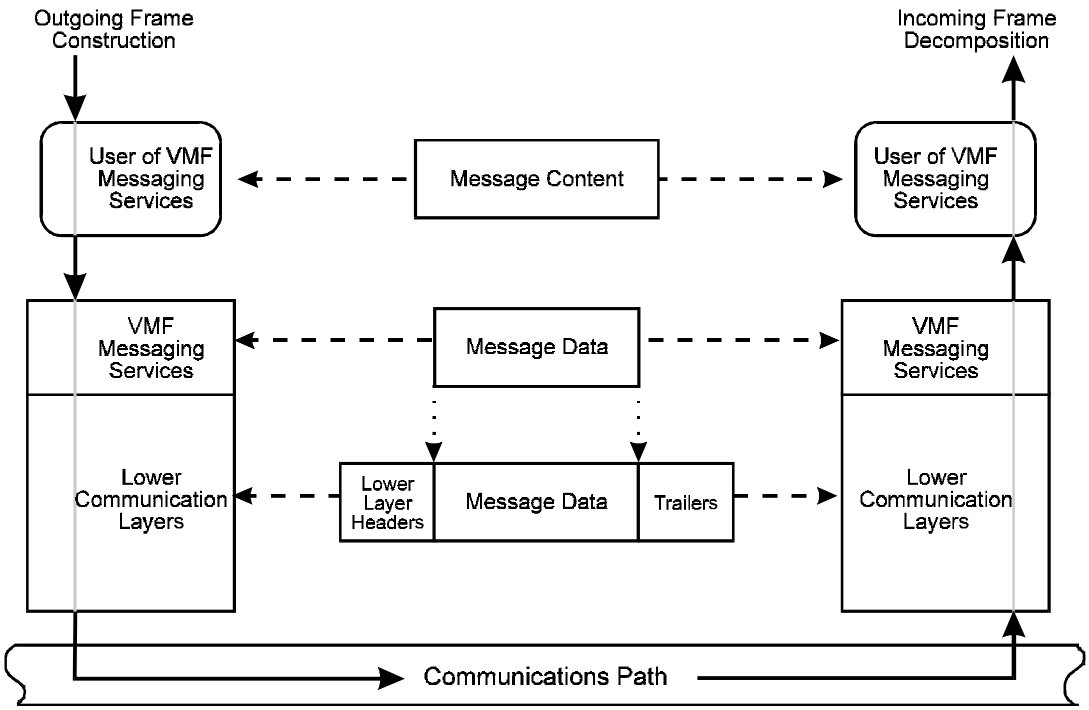


FIGURE G-2. VMF message services interaction with other communication layers.

The format of the Message Data is defined in terms of the actual data buffer or data stream used to exchange the Message Data between the VMF Messaging Services and the lower communication layers. The rationale for using the Message Data's data buffer/stream to define the format is: 1) for consistency with industry standard commercial communications hardware and software (e.g., UNIX implementations of TCP/IP), which exchange data with other software when sending or receiving as a buffer or stream of octets; 2) to provide a definition independent of the specifics of any other communication layer, consistent with the OSI ISO model principle of making communication layers independent; and 3) to avoid differences in the bit representations used to implement communications on different media. For example, on Ethernet LAN media each octet is sent least significant bit first, but on FDDI media each octet is sent most significant bit first. To achieve a universal definition of the Message Data format, its representation is defined independent of the other communication layers. The relationship of the Message Data's data buffer/stream to the VMF Messaging Services is depicted in Figure G-3. The Message Data is defined as a buffer or stream of octets. The rational for treating the Message Data as a series of octets is for consistency with the way communications data is handled by industry standard commercial communications hardware and software and for independence from platform-dependent byte ordering issues.

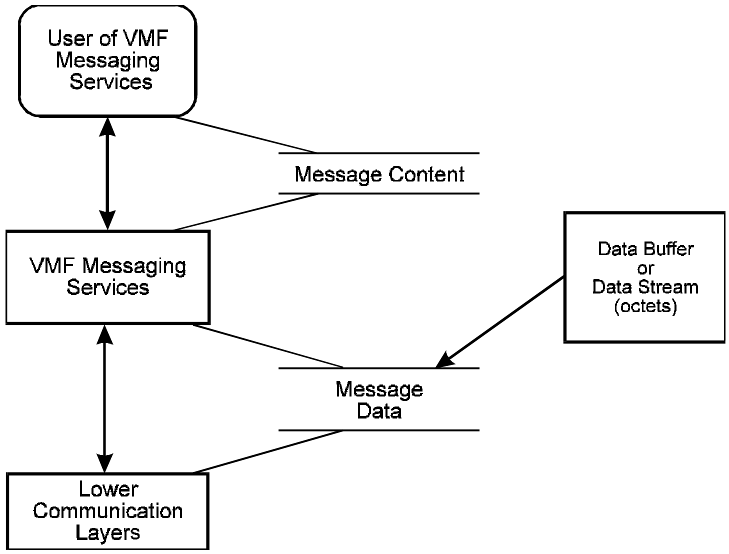


FIGURE G-3. Exchange of message data between communication layers.

G.3.1.1 Example of VMF Message Data Construction. The construction of VMF Message Data is illustrated by the example in Table G-1. The first four columns of the table provide a description of each field in the example, the field length in bits, and the value of the field in both decimal and binary representations. The last three columns show the physical encoding of the VMF Message Data. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bit(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the next least significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed. When a field has groups, the field encoding procedure is performed starting with the first group, and repeated for each successive group and individual octet, in order, until the encoding of the field is completed. The Target Number field illustrates the encoding of a field with groups. Note the LSB of a field or octet is defined as the bit having the weight of 2^0 when the field or octet is represented as a numeric value. X's are used to identify bits that are not associated with the field being encoded. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The seventh column, Octet Value – Hexadecimal, represents the octet value in hexadecimal. The last column, Octet Number, numbers the octets from first to last starting with 0.

When all fields have been encoded, any remaining unencoded bits in the last octet are filled with zeroes (zero padded). Each VMF Message is individually encoded and zero padded.

TABLE G-1. Example of VMF Message Data Construction.

Field Name	Length (bits)	Value (Dec)	Value (Bin)	Field Fragments	Octet Value (Bin)	Octet Value (hex)	Octet Number
			MSB **LSB** 2^n 2^0	2^7 2^0	2^7 2^0		
Check Fire Type	3	0	*0*00	xxxxx*0*00			
Check Fire/Cancel Check fire command	3	1	*0*01	xx*0*01xxx			
FPI	1	1	1	x1xxxxxx			
Target Number (Group 1)	7	65 (A)	*1*000001	1xxxxxxx xx*1*00000	11*0*01*0*00	C8	0
(Group 2)	7	66 (B)	*1*000010	10xxxxxx xxx*1*0000	10*1*00000	A0	1
(Group 3)	14	1543	*0*0011000000111	111xxxxx 11000000 xxxxx*0*00	111*1*0000 11000000	F0 C0	2 3
FPI (Observer URN)	1	0	0	xxxx0xxx			
FPI (First Unit URN)	1	0	0	xxx0xxxx			
GPI DTG	1	0	0	xx0xxxxx			
FPI (launcher message)	1	0	0	x0xxxxxx			
(Zero Padding)	1	0	0	0xxxxxxx	00000*0*00	00	4

Figure G-4 illustrates the octets arranged in a serial format as they would appear at the physical layer, with LSB first.

Octet 0	**Octet 1**	**Octet 2**	**Octet 3**	**Octet 4**
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
00*0*10*0*11	00000*1*01	0000*1*111	00000011	00*0*00000

FIGURE G-4. Serial representation of PDU.

G.3.2 Application Layer Data Exchange. The relationship of the Application Layer to other communication layers is shown in Figure G-5. A layered communication model is used in this example for consistency with the principles of the ISO OSI reference model. The model discussed here is tailored to focus attention specifically on the Application Layer, and the data it produces. A user of the Application Layer exchanges a VMF message with its peer at another node by sending and receiving the VMF message via the Application Layer. The Application Layer sends and receives the VMF message transparently by producing and exchanging an Application Layer Protocol Data Unit (PDU) with its peer at another node. The Application Layer PDU consists of the Application Header concatenated with the VMF message, and is sent and received via lower communication layers. The lower communication layers send and receive the VMF message transparently over a variety of communications media.

The format of the Application Layer PDU is defined in terms of the actual data buffer or data stream used to exchange the PDU between the Application Layer and the lower communication layers. The rationale for using the PDU's data buffer/stream to define the format is 1) for consistency with industry standard commercial communications hardware and software (e.g., UNIX implementations of TCP/IP), which exchange data with other software when sending or receiving as a buffer or stream of octets; 2) to provide a definition independent of the specifics of any other communication layer, consistent with the OSI model principle of making communication layers independent; and 3) to avoid differences in the bit representations used to implement communications on different media. For example, on Ethernet LAN media each octet is sent least significant bit first, but on FDDI media each octet is sent most significant bit first. To achieve a universal definition of the PDU format, its representation is defined independent of the other communication layers.

The relationship of the Application Layer PDU's data buffer/stream to the Application Layer is depicted in Figure G-6. The Application Layer PDU is defined as a buffer or stream of octets. The rational for treating the PDU as a series of octets is for consistency with the way communications data is handled by industry standard commercial communications hardware and software and for independence from platform-dependent byte ordering issues. The Application Header and the VMF message are each individually defined as a series of octets for the same reasons.

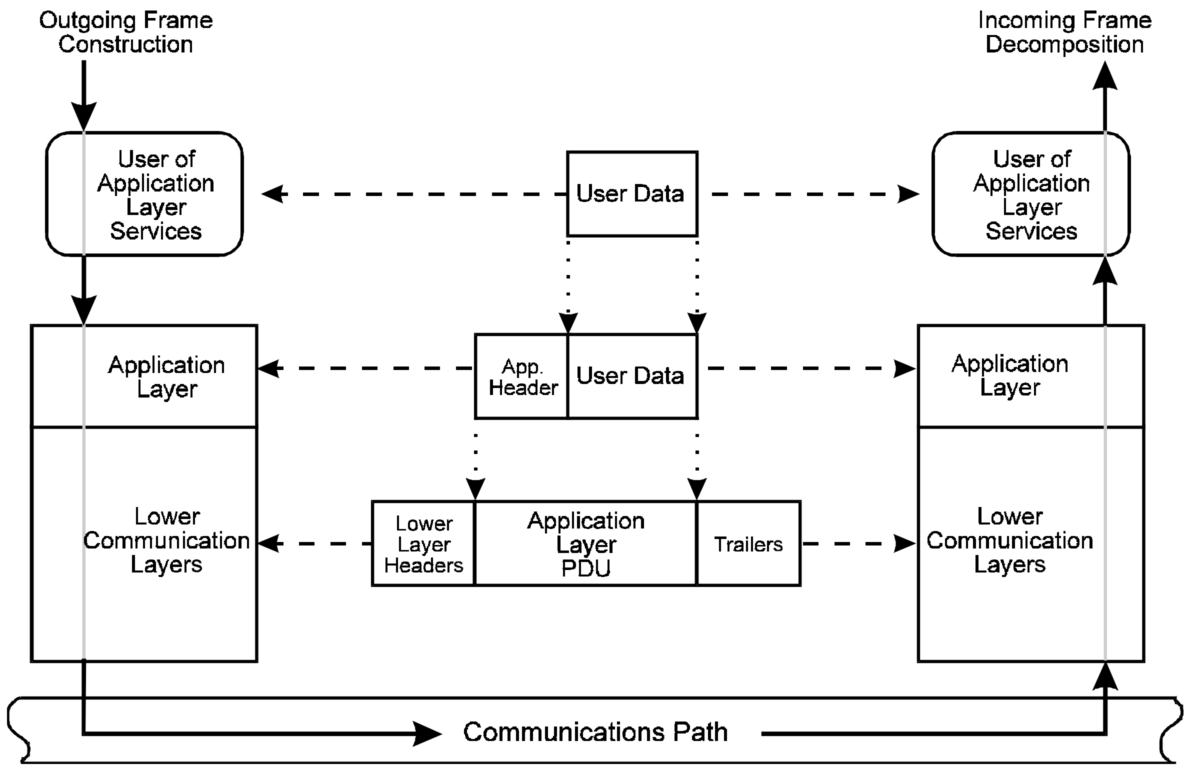


FIGURE G-5. Application layer interaction with other communication layers.

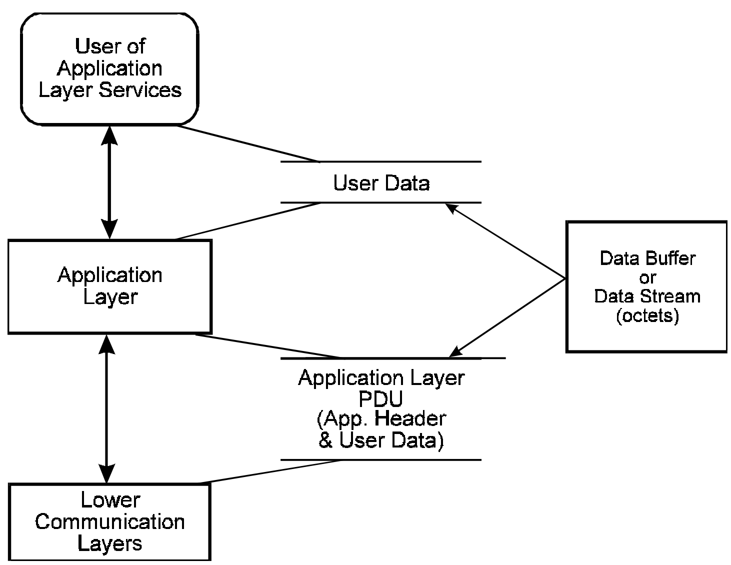


FIGURE G-6. Exchange of application layer PDU between communication layers.

G.3.2.1 Example of Application Layer PDU. The construction of an Application Layer PDU is illustrated by the example in Table G-2. The first four columns of the table provide a description of each field in the example, the field length in bits, and the value of the field in both decimal and binary representations. The last four columns show the physical encoding of the Application Layer PDU. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bit(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the next least significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed. When a field has groups, the field encoding procedure is performed starting with the first group, and repeated for each successive group and individual octet, in order, until the encoding of the field is completed. The Unit Reference Number field illustrates the encoding of a field with groups. Note the LSB of a field or octet is defined as the bit having the weight of 2^0 when the field or octet is represented as a numeric value. X's are used to identify bits that are not associated with the field being encoded. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The seventh column, Octet Value, represents the octet value in binary that should be submitted to the Transport layer. The last column, Octet Number, numbers the octets from first to last starting with 0.

When all fields have been encoded, any remaining unencoded bits in the last octet are filled with zeroes (zero padded). The Application Header is individually encoded and zero padded. The VMF message is individually encoded and zero padded before it is passed to the Application Layer to have the Application Header added.

Any of the ASCII fields (e.g. Unit Name) in the application header can be terminated by either an end of text marker, or by using the maximum number of bits. Table G-3 shows how to format the Unit Name when the Unit Name is used as part of the originator address group. The Unit Name and Unit Reference Number are mutually exclusive inside the address group – never send both, Unit Name and Unit Reference Number, in an address group. However if the address group has a Group Repeat Indicator (GRI) each of the repeatable address groups can be different address types (e.g. Unit Name or Unit Reference Number).

The Application Header is followed by the VMF message. The VMF message is shown as a single 10-octet message to complete the Application Layer PDU.

Figure G-7 provides an illustration of the Application Header as it would appear in serial form at the lower layers.

TABLE G-2. Example construction of the application header.

Syntax	**Field Description**	**Length (bits)**	**Value (Decimal)**	**Value (Binary)**	**Field Fragments**	**Octet Value (Binary)**	**OCTET Number**
				MSB ***2^n*** **LSB** **2^0**	**MSB** **2^n** **LSB** **2^0**	**MSB** **2^n** **LSB** **2^0**	
	Version	4	1	0001	x x x x 0 0 0 1		
FPI	Compression Type	1	0	0	x x x 0 x x x x		
GPI	Presence Indicator (Originator)	1	1	1	x x 1 x x x x x		
FPI	Presence Indicator (URN)	1	1	1	x 1 x x x x x x		
	Unit Reference Number (Originator)	24	23	000000000000000000010111	1 x x x x x x x 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 x 0 0 0 0 0 0 0	1 1 1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0	0 1 2
FPI	Presence Indicator (Unit Name)	1	0	0	0 x x x x x x x	0 0 0 0 0 0 0 0	3
GPI	Presence Indicator (Recipient)	1	1	1	x x x x x x x 1		
GRI	Group Repeat Indicator (Recipient)	1	0	0	x x x x x x 0 x		
FPI	Presence Indicator (URN)	1	1	1	x x x x x 1 x x		
	Unit Reference Number (Recipient URN)	24	124	000000000000000001111100	1 1 1 0 0 x x x 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 x x x x x 0 0 0	1 1 1 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0	4 5 6
FPI	Presence Indicator (Unit Name)	1	0	0	x x x x 0 x x x		
GPI	Group Presence Indicator (Information)	1	0	0	x x x 0 x x x x		
GRI	Group Repeat Indicator (Message)	1	0	0	x x 0 x x x x x		
	User Message Format	4	2	0010	1 0 x x x x x x x x x x x x 0 0	1 0 0 0 0 0 0 0	7
GPI	Group Presence Indicator (Message Identification)	1	1	1	x x x x x 1 x x		
	Functional Area Designator	4	2	0010	x 0 0 1 0 x x x		
	Message Number	7	1	0000001	1 x x x x x x x x x 0 0 0 0 0 0	1 0 0 1 0 1 0 0	8
FPI	Presence Indicator (Message Subtype #)	1	0	0	x 0 x x x x x x		
FPI	Presence Indicator (File Name)	1	0	0	0 x x x x x x x	0 0 0 0 0 0 0 0	9

TABLE G-2. Example construction of the application header.

Syntax	Field Description	Length (bits)	Value (Decimal)	Value (Binary)	Field Fragments	Octet Value (Binary)	OCTET Number
				MSB ***2^n*** **LSB** **2^0**	**MSB** **2^n** **LSB** **2^0**	**MSB** **2^n** **LSB** **2^0**	
FPI	Presence Indicator (Message Size)	1	0	0	x x x x x x x 0		
	Operation Indicator	2	0	*0*0	*x* x x x x *0* 0 x		
	Retransmit Indicator	1	0	0	x x x x 0 x x x		
	Message Precedence Code	3	7	*1*11	x *1* 1 1 x x x x		
	Security Classification	2	0	*0*0	0 x x x x x x x x x x x x x x *0*	0 *1* 1 1 0 *0* 0 0	10
FPI	FPI for Control/Release Marking	1	0	0	x x x x x x 0 x		
GPI	GPI for Originator DTG	1	1	1	x x x x x 1 x x		
	Year	7	96	*1*100000	0 0 0 0 0 x x x x x x x x x *1* 1	0 0 0 0 0 1 0 *0*	11
	Month	4	7	*0*111	x x *0* 1 1 1 x x		
	Day	5	1	*0*0001	0 1 x x x x x x x x x x x *0* 0 0	0 1 *0* 1 1 1 *1* 1	12
	Hour	5	8	*0*1000	*0* 1 0 0 0 x x x	*0* 1 0 0 0 *0* 0 0	13
	Minute	6	32	*1*00000	x x *1* 0 0 0 0 0		
	Second	6	16	*0*10000	0 0 x x x x x x x x x x *0* 1 0 0	0 0 *1* 0 0 0 0 0	14
FPI	DTG Extension	1	0	0	x x x 0 x x x x		
GPI	GPI for Perishability DTG	1	0	0	x x 0 x x x x x		
GPI	GPI for ACK Request Group	1	0	0	x 0 x x x x x x		
GPI	GPI for Response Data Group	1	0	0	0 x x x x x x x	0 0 0 0 *0* 1 0 0	15
GPI	GPI for Reference Message Data	1	0	0	x x x x x x x 0		
	(Zero Padding)	7	0	*0*000000	*0* 0 0 0 0 0 0 x	*0* 0 0 0 0 0 0 0	16

TABLE G-3. Example of a Unit Name as Originator.

Syntax	Field Description	Length (bits)	Value (Decimal)	Value (Binary)	Field Fragments	Octet Value (Binary)	OCTET Number
				MSB 2^n **LSB** 2^0	**MSB** 2^n **LSB** 2^0	**MSB** 2^n **LSB** 2^0	
	Version	4	1	*0*001	x x x x *0* 0 0 1		
FPI	Compression Type	1	0	0	x x x 0 x x x x		
GPI	Presence Indicator (Originator)	1	1	1	x x 1 x x x x x		
FPI	Presence Indicator (URN)	1	0	0	x 0 x x x x x x		
FPI	Presence Indicator (Unit Name)	1	1	1	1 x x x x x x x	1 0 0 0 0 0 0 0	0
	Unit Name (Originator)	448Max	"UNITA"				
	"U"	7	85	*1*010101	x *1* 0 1 0 1 0 1		
	"N"	7	78	*1*001110	0 x x x x x x x x x *1* 0 0 1 1 1	0 *1* 0 1 0 1 0 1	1
	"I"	7	73	*1*001001	0 1 x x x x x x x x x *1* 0 0 1 0	0 1 *1* 0 0 1 1 1	2
	"T"	7	84	*1*010100	1 0 0 x x x x x x x x x *1* 0 1 0	1 0 0 *1* 0 0 1 0	3
	"A"	7	65	*1*000001	0 0 0 1 x x x x x x x x x *1* 0 0	0 0 0 1 *1* 0 1 0	4
	End of text marker (ANSI ASCII DEL)	7	127	*1*111111	1 1 1 1 1 x x x x x x x x x *1* 1	1 1 1 1 1 *1* 0 0	
GPI	Presence Indicator (Recipient)	1	1	1	x x x x x 1 x x		
encode rest of the message as in Figure G-3							

Octet 0	Octet 1	Octet 2	Octet 11	Octet 12	Octet 13	Octet 14	Octet 15	Octet 16
2^0 2^7	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**
1 0 0 *0* 0 1 1 1	1 1 0 1 0 0 0 0	0 0 0 0 0 0 0 0	*0* 0 1 0 0 0 0 0	1 *1* 1 1 1 *0* 1	0 0 *0* 0 0 0 1 *0*	0 0 0 0 0 *1* 0 0	0 0 1 *0* 0 0 0 0	0 0 0 0 0 0 0 *0*

FIGURE G-7. Application header (octets).

G.3.3 Transport Layer Data Exchange. The relationship of the Transport Layer to other communication layers is shown in Figure G-8. A user of the Transport Layer exchanges data with its peer at another node by sending and receiving the Application Layer PDU via the Transport Layer. The Transport Layer sends and receives the Application Layer PDU transparently by producing and exchanging a Transport Layer Protocol Data Unit (PDU) with its peer at another node. The Transport Layer PDU consists of the Transport Header concatenated with the Application Layer PDU, and is sent and received via lower layer communication layers. The lower communication layers send and receive the Transport PDU transparently over a variety of communications media.

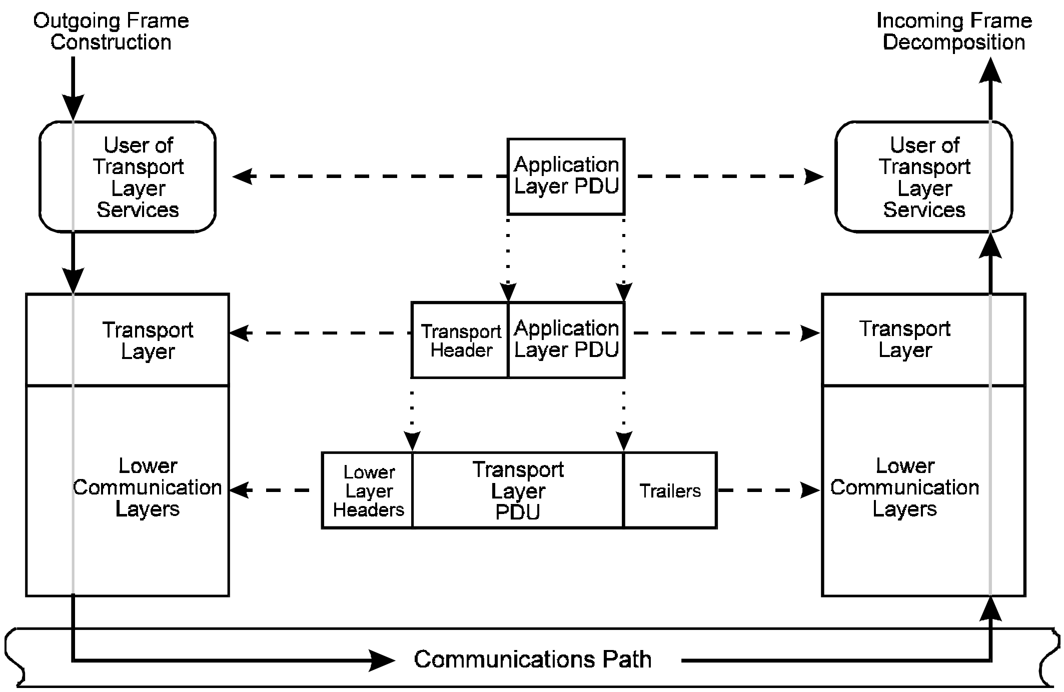


FIGURE G-8. Transport layer interaction with other communication layers.

The relationship of the Transport Layer PDU's data buffer/stream to the Application Layer is depicted in Figure G-9. The Transport Layer PDU is defined as a buffer or stream of octets consisting of the VMF message, Application Header and Transport Header.

G.3.3.1 An Example of UDP Header Construction. UDP is described by RFC 768. The UDP header from RFC 768 consists of 8 octets as shown in Figure G-10 with the example values to be used for this appendix. . Since the RFC treats bit 0 as most significant bit (MSB), Figures G-10 and G-11 show B_0 as MSB. For this example, the source has a value of 1581, destination of 1581, length of 30 and the checksum equals 3491. MIL-STD-188-220 typically treats the least significant bit as bit 0.

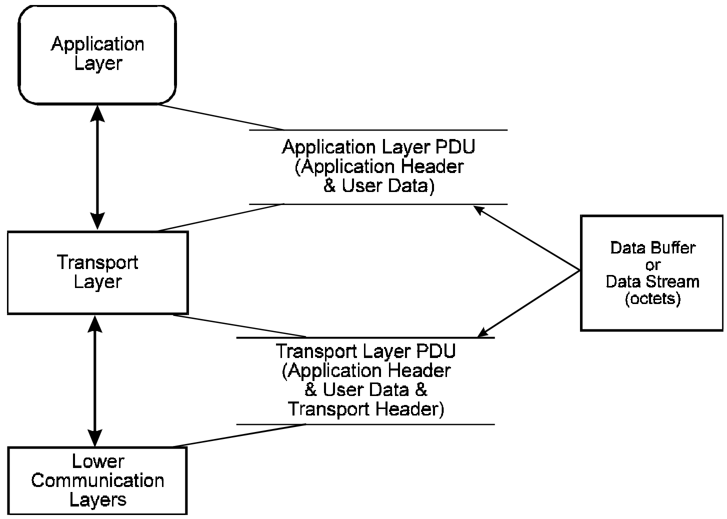


FIGURE G-9. Exchange of transport layer PDU between communication layers.

0 7	**8 15**	**16 23**	**24 31**
UDP Source (1581)		UDP Destination (1581)	
UDP Length (30)		UDP Checksum (3491)	

Bit 0 is most significant bit (MSB)

FIGURE G-10. UDP header.

Figure G-11 illustrates the eight octets comprising UDP with the binary bit patterns. Each octet is marked to show both the MSB and LSB of each octet. It demonstrates how each of the octets are arranged and passed in order to next layer.

Octet 0	**Octet 1**	**Octet 2**	**Octet 3**
B_0 B_7	B_8 B_{15}	B_{16} B_{23}	B_{24} B_{31}
2^7 2^0	2^7 2^0	2^7 2^0	2^7 2^0
00000110	00101101	00000110	00101101
UDP Source (1581)		**UDP Destination (1581)**	

Octet 4	**Octet 5**	**Octet 6**	**Octet 7**
B_0 B_7	B_8 B_{15}	B_{16} B_{23}	B_{24} B_{31}
2^7 2^0	2^7 2^0	2^7 2^0	2^7 2^0
00000000	00011110	00001101	10100011
UDP Length (30)		**UDP Checksum (3491)**	

FIGURE G-11 Octet representation of UDP header.

The construction of a Transport Layer Header is illustrated by the example in Table G-4. The first four columns of the table provide a description of each field in both decimal and binary representations. The last two columns show the physical encoding of the Transport Layer PDU. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bits(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the Next last significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed.. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The last column, Octet Number, numbers the octets from first to last starting with 0.

TABLE G-4. Example construction of UDP header.

Field Name	**Length**	**Value (Dec)**	**Value (Bin)**	**Field Fragments**	**Octet Value (Bin)**	**Octet Number**
			MSB **LSB** ***2^{15}*** **2^0**	**2^7** **2^0**	**2^7** **2^0**	
UDP Source	16	1581	*0*000011000101101	*0* 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1	*0* 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1	0 1
UDP Destination	16	1581	*0*000011000101101	*0* 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1	*0* 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1	2 3
UDP Length	16	30	*0*000000000011110	*0* 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0	*0* 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0	4 5
UDP Checksum	16	3491	*0*000000010100011	*0* 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1	*0* 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1	6 7

Table G-5 illustrates the eight octets of the Transport Header showing the binary value of the octet, the octet number (0-7) and the field represented by each octet. Note that the bit with the bold italicized font identifies the MSB (2^n) of the field, not the octet.

Figure G-12 provides a serial representation of the UDP header as it would appear at the physical layer.

TABLE G-5. Octet representation of UDP header.

Octet Value (Binary)	**Octet Number**	**Field Name**
2^7 2^0		
0 0 0 0 0 1 1 0	0	**Source**
0 0 1 0 1 1 0 1	1	**Source**
0 0 0 0 0 1 1 0	2	**Destination**
0 0 1 0 1 1 0 1	3	**Destination**
0 0 0 0 0 0 0 0	4	**Length**
0 0 0 1 1 1 1 0	5	**Length**
0 0 0 0 1 1 0 1	6	**Checksum**
1 0 1 0 0 0 1 1	7	**Checksum**

Octet 0	**Octet 1**	**Octet 2**	**Octet 3**	**Octet 4**	**Octet 5**	**Octet 6**	**Octet 7**
2^0 2^7	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**
0 1 1 0 0 0 0 *0*	1 0 1 1 0 1 0 0	0 1 1 0 0 0 0 *0*	1 0 1 1 0 1 0 0	0 0 0 0 0 0 0 *0*	0 1 1 1 1 0 0 0	1 0 1 1 0 0 0 *0*	1 1 0 0 0 1 0 1

FIGURE G-12. Serial representation of UDP header.

G.3.4 Network Layer Data Exchange. The relationship of the Network Layer to other communication layers is shown in Figure G-13. A user of the Network Layer exchanges data with its peer at another node by sending and receiving the Transport Layer PDUs via the Network Layer. The Network Layer sends and receives the Transport Layer PDUs transparently by producing and exchanging a Network Layer PDU. The Network Layer PDU consists of the Network Headers concatenated with the Transport Layer PDU, and is sent and received via lower layer communication layers. The lower communication layers send and receive the Network Layer PDU transparently over a variety of communications media.

The relationship of the Network Layer PDU's data buffer/stream to the Transport Layer is depicted in Figure G-14. The Network Layer PDU is defined as a buffer or stream of octets consisting of the VMF message, Application Header, Transport Header and Network Headers. There are two Network Headers in the Network Layer PDU when using MIL-STD-188-220.

The Internet Protocol (IP) is described by RFC 791. The IP header from RFC 791 is shown in Figure G-15 with the example values to be used for this appendix.

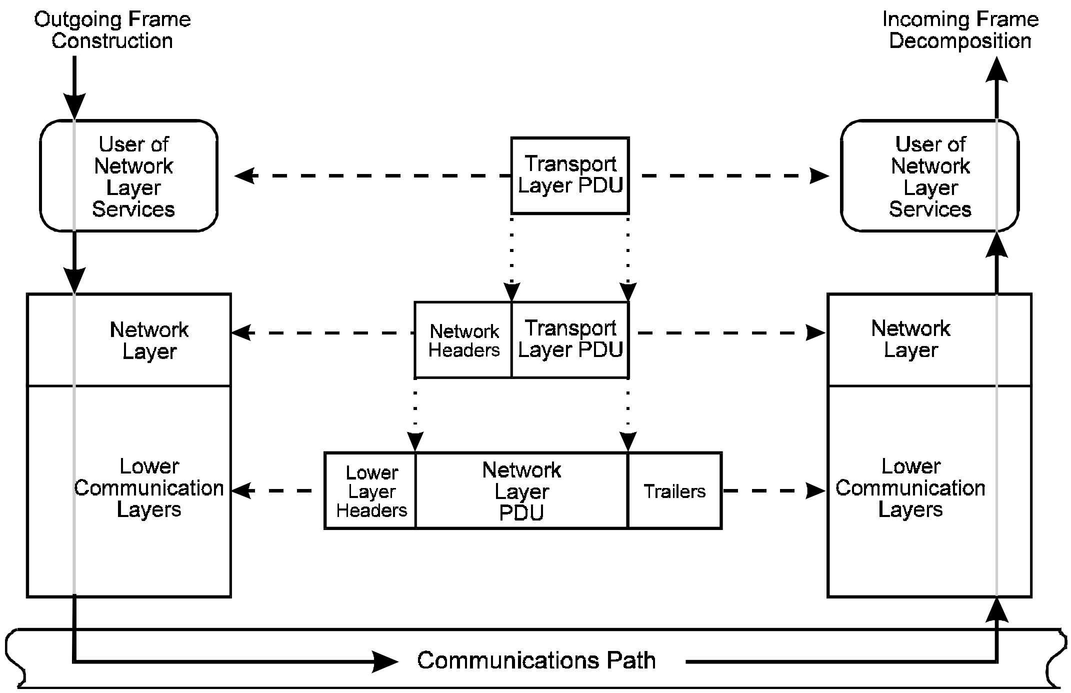


FIGURE G-13. Network layer interaction with other communication layers.

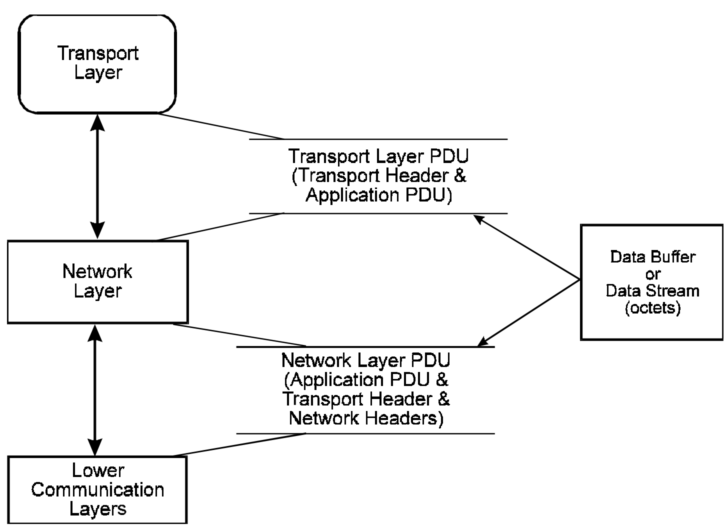


FIGURE G-14. Exchange of network layer PDU between communication layers.

<table>
<tr><td colspan="2">0
0 1 2 3 4 5 6 7</td><td>1
8 9 0 1 2 3 4 5</td><td colspan="2">2
6 7 8 9 0 1 2 3</td><td>3
4 5 6 7 8 9 0 1</td></tr>
<tr><td>Ver (4)</td><td>IHL (5)</td><td>Type of Service (0)</td><td colspan="3">Total Length (50)</td></tr>
<tr><td colspan="3">Identification (1)</td><td>Flag (0)</td><td colspan="2">Fragment Offset (0)</td></tr>
<tr><td colspan="2">Time to Live (50)</td><td>Protocol (17)</td><td colspan="3">Header Checksum (4093)</td></tr>
<tr><td colspan="6">Source Address (192.31.124.65)</td></tr>
<tr><td colspan="6">Destination Address (192.31.124.61)</td></tr>
</table>

FIGURE G-15. IP header.

G.3.4.1 Example of Internet Layer Header. The construction of an Internet Layer Header is illustrated by the example in Table G-6. The first four columns of the table provide a description of each field in the example, the field length in bits, and the value of the field in both decimal and binary representations. The last three columns show the physical encoding of the Internet Layer Header. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bit(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the next least significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed. X's are used to identify bits that are not associated with the field being encoded. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The last column, Octet Number, numbers the octets from first to last starting with 0.

Figure G-16 illustrates the Internet Header demonstrating the relationship between the individual bits (B^0 - B^7), the bit weighting (2^7 - 2^0), the individual fields and the example bit patterns associated with each field.

TABLE G-6. Example construction of IP header.

Field Name	**Length**	**Value (Dec)**	**Value (Binary)** 2^n 2^0	**Field Fragments** 2^n 2^0	**Octet Value (Binary)** 2^n 2^0	**Octet Number**
Version	4	4	*0*100	*0*100xxxx		
Internet Header Length	4	5	*0*101	xxxx*0*101	*0*100*0*101	0
Type of Service	8	0	*0*0000000	*0*0000000	*0*0000000	1
Length	16	50	*0*000000000110010	*0*0000000 00110010	*0*0000000 00110010	2 3
Identification	16	111	*0*000000000000001	*0*0000000 00000001	*0*0000000 00000001	4 5
Flags	3	0	*0*00	*0*00xxxxx		
Fragmentation Offset	13	0	*0*000000000000	xxx*0*0000 00000000	000*0*0000 00000000	6 7
Time to Live	8	50	*0*0110010	*0*0110010	*0*0110010	8
Protocol	8	17	*0*0010001	*0*0010001	*0*0010001	9
Header Checksum	16	4093	*0*000111111111101	*0*0001111 11111101	*0*0001111 11111101	10 11
Source Address	32	192.31.124.65	*1*100000000011111 0111110001000001	*1*1000000 00011111 01111100 01000001	*1*1000000 00011111 01111100 01000001	12 13 14 15
Destination Address	32	192.31.124.61	*1*100000000011111 0111110000111101	*1*1000000 00011111 01111100 00111101	*1*1000000 00011111 01111100 00111101	16 17 18 19

Octet 0		**Octet 1**	**Octet 2**		**Octet 3**
B_0 B_7		B_0 B_7	B_0 B_7		B_0 B_7
2^7 2^0		2^7 2^0	2^7 2^0		2^7 2^0
0 1 0 0 *0* 1 0 1		*0* 0 0 0 0 0 0 0	*0* 0 0 0 0 0 0 0		0 0 1 1 0 0 1 0
Ver (4)	**IHL (5)**	**Type of Service (0)**	**Total Length (50)**		

Octet 4	**Octet 5**	**Octet 6**		**Octet 7**
B_0 B_7	B_0 B_7	B_0 B_7		B_0 B_7
2^7 2^0	2^7 2^0	2^7 2^0		2^7 2^0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 1	0 0 0	*0* 0 0 0 0	0 0 0 0 0 0 0 0
Identification (1)		**Flag (0)**	**Fragment Offset (0)**	

Octet 8	**Octet 9**	**Octet 10**	**Octet 11**
B_0 B_7	B_0 B_7	B_0 B_7	B_0 B_7
2^7 2^0	2^7 2^0	2^7 2^0	2^7 2^0
0 0 1 1 0 0 1 0	*0* 0 0 1 0 0 0 1	*0* 0 0 0 1 1 1 1	1 1 1 1 1 1 0 1
Time (50)	**Protocol (17)**	**Header Checksum (4093)**	

Octet 12	**Octet 13**	**Octet 14**	**Octet 15**
B_0 B_7	B_0 B_7	B_0 B_7	B_0 B_7
2^7 2^0	2^7 2^0	2^7 2^0	2^7 2^0
1 1 0 0 0 0 0 0	0 0 0 1 1 1 1 1	0 1 1 1 1 1 0 0	0 1 0 0 0 0 0 1
Source Address (192.31.124.65)			

Octet 16	**Octet 17**	**Octet 18**	**Octet 19**
B_0 B_7	B_0 B_7	B_0 B_7	B_0 B_7
2^7 2^0	2^7 2^0	2^7 2^0	2^7 2^0
1 1 0 0 0 0 0 0	0 0 0 1 1 1 1 1	0 1 1 1 1 1 0 0	0 0 1 1 1 1 0 1
Destination Address (192.31.124.61)			

FIGURE G-16. Octet representation of IP header.

G.3.4.2 Example of Intranet Layer Header. The construction of an Intranet Layer Header is illustrated by the example in Table G-7. The first four columns of the table provide a description of each field in the example, the field length in bits, and the value of the field in both decimal and binary representations. The last three columns show the physical encoding of the Intranet Layer Header. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bit(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the next least significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed. X's are used to identify bits that are not associated with the field being encoded. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The last column, Octet Number, numbers the octets from first to last starting with 0. This example only illustrates the Intranet Header fields that must be transmitted as a minimum.

TABLE G-7. Example construction of Intranet header (minimum).

Field Name	**Length**	**Value (Decimal)**	**Value & Byte Representation (Binary)** 2^n … 2^0	**Field Fragments** 2^7 … 2^0	**Octet Value (Binary)** 2^7 … 2^0	**Octet Number**
Version Number	4	0	*0*000	xxxx*0*000		
Message Type	4	4	*0*100	*0*100xxxx	*0*100*0*000	0
Intranet Header Length	8	3	*0*0000011	*0*0000011	*0*0000011	1
Type of Service	8	0	*0*0000000	*0*0000000	*0*0000000	2

The Intranet layer is defined in MIL-STD-188-220 and is shown in Figure G-17 with the example values used in this appendix.

<table>
<tr><th colspan="2">Octet 0</th><th>Octet 1</th><th>Octet 2</th></tr>
<tr><td colspan="2">2^0 … 2^7</td><td>2^0 … 2^7</td><td>2^0 … 2^7</td></tr>
<tr><td>0 0 0 *0*</td><td>0 0 1 *0*</td><td>1 1 0 0 0 0 0 *0*</td><td>0 0 0 0 0 0 0 *0*</td></tr>
<tr><td>Version (0)</td><td>Message Type (4)</td><td>Intranet Header Length (3)</td><td>Type of Service (0)</td></tr>
</table>

FIGURE G-17. Intranet header .

Figure G-18 provides a serial representation of the Network Layer PDU as it would appear at the physical layer.

Intranet header			**IP header**							
Octet 0	**Octet 1**	**Octet 2**	**Octet 3**	**Octet 4**	**Octet 5**	**Octet 6**	**Octet 7**	**Octet 8**	**Octet 9**	**Octet 10**
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
000*00*01*0*	1100000*0*	0000000*0*	101*00*01*0*	0000000*0*	0000000*0*	01001100	0000000*0*	10000000	0000000*0*	00000000

IP header (continued)										
Octet 11	**Octet 12**	**Octet 13**	**Octet 14**	**Octet 15**	**Octet 16**	**Octet 17**	**Octet 18**	**Octet 19**	**Octet 20**	**Octet 21**
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
0100110*0*	1000100*0*	1111000*0*	10111111	0000001*1*	11111000	00111110	10000010	0000001*1*	11111000	00111110

IP header (end)	
Octet 22	
2^0	2^7
10111100	

FIGURE G-18. Serial representation of network layer PDU.

G.3.6 Data Link Layer Data Exchange. The relationship of the Data Link Layer to other communication layers is shown in Figure G-19. A user of the Data Link Layer exchanges the Network Layer PDU with its peer at another node by sending and receiving the Network PDU via the Data Link Layer. The Data Link Layer sends and receives the VMF message transparently by producing and exchanging a Data Link Layer PDU with its peer at another node. The Data Link Layer PDU consists of the Transmission Header, and Data Link Frame Header, Network PDU, and the Data Link Frame Trailer, and is sent and received via the Physical layer. The Physical layer sends and receives the VMF message transparently over a variety of communications media.

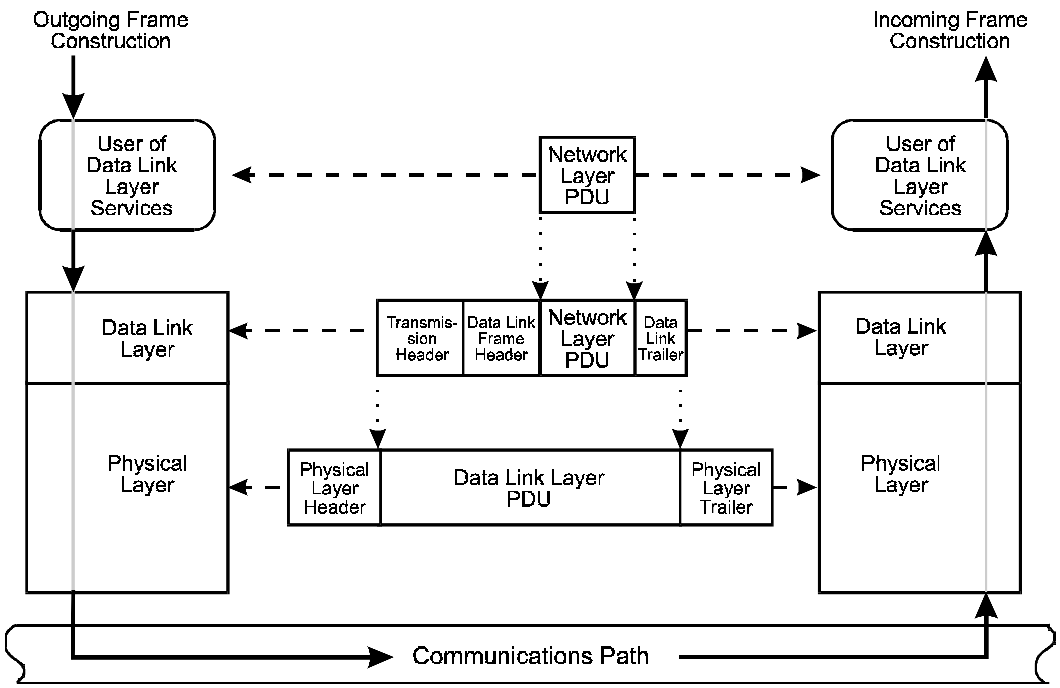


Figure G-19. Data link layer interaction with other communication layers.

The format of the Data Link Layer PDU is defined in terms of the actual data buffer or data stream used to exchange the PDU between the Network Layer and the Physical Layer. The relationship of the Data Link Layer PDU's data buffer/stream to the Intranet Layer is depicted in Figure G-20. The Data Link Layer PDU is defined as a buffer or stream of octets consisting of the Transmission Header, Data Link Frame Header, Network PDU and Data Link Layer trailer.

G.3.6.1 Example of Data Link Layer PDU. The Data Link Layer PDU consists of the Transmission Header, Data Link Frame Header, Followed by the information field and Data Link Frame Trailer as shown in Figure G-21. The information field consists of the Network PDU described previously (VMF message, Application Header, Transport Header, IP Header and Intranet Header).

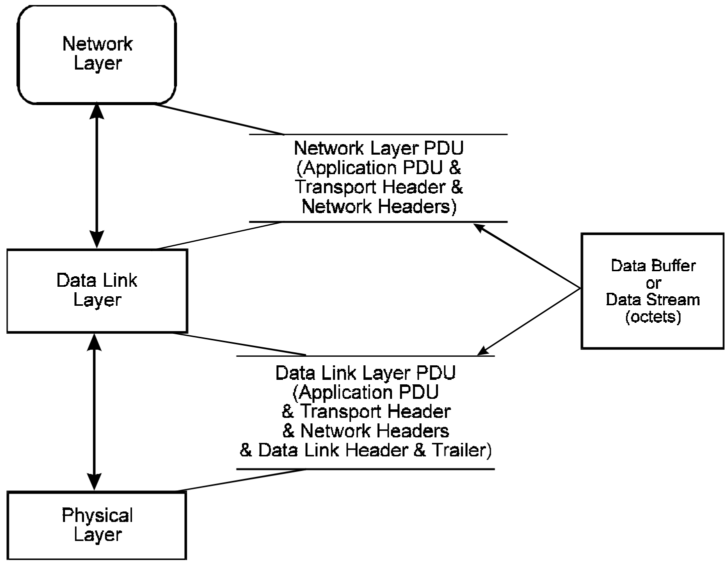


FIGURE G-20. Exchange of data link layer PDU between communication layers.

Trans-mission Header	Data Link Frame Header	Information Field	Data Link Frame Trailer

FIGURE G-21. Data link layer PDU.

Table G-8 illustrates the Data Link Frame Header, and Table G-9 illustrates the Data Link Frame Trailer. The first four columns of the tables provide a description of each field in the example, the field length in bits, and the value of the field in both decimal and binary representations. The last three columns show the physical encoding of the Data Link Frame. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bit(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the next least significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The last column, Octet Number, numbers the octets from first to last starting with 0.

TABLE G-8. Example construction of data link frame header.

Field Name	**Length**	**Value (Dec)**	**Value (Binary)** 2^n … 2^0	**Field Fragments**	**Octet Value (Binary)** 2^n … 2^0	**Octet Number**
Flag	8	126	*0*1111110	*0*1111110	*0*1111110	0
Command/Response Bit	1	0	0	xxxxxxxx0		
Source Address	7	7	*0*000111	*0*000111x	*0*0001110	1
Extension Bit	1	1	1	xxxxxxx1		
Destination Address	7	4	*0*000100	*0*000100x	*0*0001001	2
Control Field	8	19	*0*0010011	*0*0010011	*0*0010011	3

TABLE G-9. Example construction of data link frame trailer.

Frame Check Sequence (transmitted MSB first)	32	162159487	*0*0001001101010100101110101111111	*0*0001001 10101010 01011101 01111111	*0*0001001 10101010 01011101 01111111	0 1 2 3
Flag	8	126	*0*1111110	*0*1111110	*0*1111110	4

Table G-10 illustrates the octets comprising the Data Link Frame showing the actual bit patterns from the previous examples for each layer, the octet number based on each individual layer, and the octet number based on entire Data Link Frame. This data is shown in serial representation as it would be transmitted in Figure G-22.

TABLE G-10. Octets comprising data link frame.

2^7 2^0	Nomenclature	Octet Number (Individual Layer)	Octet Number (Entire Transaction)
01111110	**Flag**	0	0
00001110	**Source Address**	1	1
00001001	**Destination Address**	2	2
00010011	**Control Field**	3	3
01000000		0	4
00000011	***INTRANET HEADER***	1	5
00000000		2	6
01000101		0	7
00000000		1	8
00000000		2	9
00110010		3	10
00000000		4	11
00000001		5	12
00000000		6	13
•	***IP HEADER***	•	•
•		•	•
01111100		18	25
00111101		19	26
00000110		0	27
00101101		1	28
00000110		2	29
•	***UDP HEADER***	•	•
•		•	•
10100011		7	34

TABLE G-10. Octets comprising data link frame.

2^7 2^0	Nomenclature	Octet Number (Individual Layer)	Octet Number (Entire Transaction)
1110*0*001		0	35
00001011		1	36
•		•	•
•		•	•
0*1*110000		10	45
0000010*0*		11	46
01*0*111*1*1	***APPLICATION HEADER***	12	47
01000*0*00		13	48
•		•	•
•		•	•
0000*0*100		15	50
*0*0000000		16	51
11*0*01*0*00		0	52
10*1*00000		1	53
111*1*0000		2	54
11000000		3	55
•	***CHECKFIRE MESSAGE***	•	•
•		•	•
00000*0*00		4	56
*0*0001001	**Note: FCS transmitted MSB First**	0	57
10101010	**FCS**	1	58
01011101		2	59
01111111		3	60
*0*1111110	**Flag**	0	61

DATA LINK FRAME HEADER				INTRANET HEADER				IP									
0	1	2	3	4		5	6	7		8	9	10	11	12	13		14
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0	2^7	2^0 2^7	2^0 2^7	2^0	2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0	2^7	2^0 2^7
FLAG	SRC	DST	CNTL	V	T	LEN	TOS	L	V	TOS	Total Length		Identification		Offset	flag	Offset
01111110	01110000	10010000	11001000	0000	0010	11000000	00000000	1010	0010	00000000	00000000	01001100	00000000	10000000	00000	000	00000000

IP		UDP			
25	26	27	28	29	30
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
DESTINATION		SOURCE		DESTINATION	
00111110	10111100	01100000	10110100	01100000	10110100

APP. HEADER		
34	35	36
2^0 2^7	2^0 2^7	2^0 2^7
CHKSM	GPI-FPI-ORIG	
11000101	10000111	11010000

APP. HEADER			
45	46	47	48
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
Message Size- etc.			GPI-YR
00001110	00100000	11111010	00000010

APP. HEADER		VMF MESSAGE			
50	51	52	53	54	55
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
Min/Sec-etc.		CF-etc.	GROUP-etc		
00100000	00000000	00010011	00000101	00001111	00000011

LINK FRAME TRAILER					
56	57	58	59	60	61
2^0 2^7	2^7 2^0	2^7 2^0	2^7 2^0	2^7 2^0	2^0 2^7
Pad	FCS				FLAG
00000000	00001001	10101010	01011101	01111111	01111110

FIGURE G-22. Serial representation of data link layer PDU.

G.3.6.1.1 Zero Bit Insert/v36 scramble/FEC/TDC of the data link frame. The Data Link Frame must be zero inserted to prevent any part of the data accidentally being interpreted as a Frame Flag. Also in our example scrambling, FEC and TDC are being used. Figure G-23 shows some of the example data before applying zero-bit insertion, scrambling, FEC or TDC. After zero bit insertion, scrambling, FEC and TDC, the fields are not easy to identify; therefore field names are not shown.

1 word		**2 word**		**3 word**	
2^0 2^7	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**
0x7e70		0x90c8		0x02c0	

30 word		**31 word**		**32 word**	
2^0 2^7	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**
0x09aa		0x5d7f		0x7e00	

FIGURE G-23. Data before zero bit insertion in transmission order.

The following is a Hex dump of the data link frame in the different stages: (a) zero bit inserted, (b) scrambled, (c) FEC, and (d) TDC:

Note: In the following dumps the 16 bit values are in transmission order. The TWC in the physical layer is defined in words and fields are no longer easily distinguishable.

a. Data after zero bit insertion (505 bits plus 7 padding bits)
0x7e70 0x90c8 0x02c0 0x00a2 0x0000 0x4c00 0x8000 0x004c 0x88f0 0xbe81 0xf60f 0x9040 0x7d83 0xe5e3 0x05a3 0x05a0 0x03c5 0x862c 0x3e40 0x0002 0x9f00 0x0002 0x5200 0x1c41 0xf202 0x0420 0x0013 0x050f 0x0300 0x09aa 0x5d7d 0xbf00

b. Data after V.36 scrambling (512 bits)
0x8f80 0x872a 0xa161 0x7a0a 0xbfaa 0x524c 0x50c3 0x50aa 0x024c 0x6cc2 0x9ca9 0x6b17 0xe9f3 0x0403 0xbda9 0xfe4c 0xfc54 0x3014 0x02e2 0xe3a7 0xb9fa 0xdf90 0x0006 0x2754 0xf1bf 0x5f20 0x0b70 0xe695 0x59a2 0xfc47 0x616b 0x5d41

c. Data after FEC(Golay 24,12) (data size in bits: 0x0408 plus 8 padding bits)
Golay (24,12) is derived from Golay (23,12):See paragraph F 4.1 for details.
0x8f8a 0x5a08 0x7898 0x2aae 0x8616 0x140a 0x7a0b 0xf0ab 0xf3e8 0xaa54 0x7624 0xc5a0 0x50c6 0xde35 0x0622 0xaa06 0x0a24 0xc5a0 0x6cc0 0x4029 0xc884 0xa960 0x08b1 0x7c8c 0xe9f3 0x1e30 0x424a 0x03b9 0xb8da 0x9dc0 0xfe40 0x8acf 0xc6f6 0x543d 0xc201 0x49f0 0x02e8 0xa22e 0x3632 0xa7b7 0x3c9f 0xa4d0 0xdf93 0x3e00 0x0000 0x0622 0x6a75 0x4a8e 0xf1bd 0xe6f5 0xfae8 0x200f 0x68b7 0x0c9a 0xe69b 0x5e55 0x9a5c 0xa2f3 0x54c4 0x7c94 0x6169 0x9cb5 0xd5ec 0x4105 0x5c00

d. Data after TDC(16,24) (data size in bits: 0x0480)
0x8623 0x0888 0x2f7f 0x18c1 0xee2e 0x9158 0xbe20 0x8447 0xa59c 0x479f 0x6403 0x5601 0xe805 0x33f1 0xace0 0x0d10 0x6d95 0x8e88 0x0f50 0xca80 0xd4a3 0x2285 0xb2e0 0x0000 0x9c38 0x9e09 0xc861 0x5a19 0x9c58 0x0e7b 0x3cfa 0xa539 0xb4b8 0xcd81 0xa2f2 0xb268 0x3381 0x1670 0xc46b 0xb328 0x3f91 0x5712 0x25ea 0xa578 0xe82b 0x8429 0xcecb 0x0000 0xdb40 0xcda0 0xfac0 0xd440 0x0000 0x5d40 0x1a00 0xd4e0 0xce40 0x43c0 0xc380 0xcf40 0xfd80 0xb160 0x6e00 0xaae0 0xd1c0 0xee60 0xe040 0x1fa0 0x7ce0 0x8fe0 0x9800 0x0000

G.3.6.1.2 Construction of the Transmission Header. The Transmission Header precedes the data link frame and formatted as defined in Table G-11.

G.3.6.1.3 Zero Bit Insert/v36 scramble/FEC of the Transmission Header. The Transmission Header must be zero inserted to prevent any part of the data accidentally being interpreted as a Frame Flag. After zero bit insertion, the fields are not easy to identify; therefore field names are not shown. The following is a Hex dump of the Transmission Header of zero bit inserted:

Transmission Header after zero bit insertion (Size In Bits 0x0040)
0x7ee0 0x001c 0x2119 0x707e

G.3.6.1.4 Completed Data Link Layer PDU to be passed to the physical layer. The data link layer passes the Data Link Layer PDU to the physical layer The elements of a Data Link Layer PDU include one transmission header and one or more PDUs. The following complete data link PDU (consisting of transmission header and data link frame) will be passed to the physical layer:

Complete Data Link Layer PDU

a. Transmission Header:
0x7ee0 0x001c 0x2119 0x707e

b. Data Link Layer Frame (72 16 bit words):
0x8623 0x0888 0x2f7f 0x18c1 0xee2e 0x9158 0xbe20 0x8447 0xa59c 0x479f 0x6403 0x5601 0xe805 0x33f1 0xace0 0x0d10 0x6d95 0x8e88 0x0f50 0xca80 0xd4a3 0x2285 0xb2e0 0x0000 0x9c38 0x9e09 0xc861 0x5a19 0x9c58 0x0e7b 0x3cfa 0xa539 0xb4b8 0xcd81 0xa2f2 0xb268 0x3381 0x1670 0xc46b 0xb328 0x3f91 0x5712 0x25ea 0xa578 0xe82b 0x8429 0xcecb 0x0000 0xdb40 0xcda0 0xfac0 0xd440 0x0000 0x5d40 0x1a00 0xd4e0 0xce40 0x43c0 0xc380 0xcf40 0xfd80 0xb160 0x6e00 0xaae0 0xd1c0 0xee60 0xe040 0x1fa0 0x7ce0 0x8fe0 0x9800 0x0000

G.3.7 Physical Layer Data Exchange. The relationship of the Physical Layer to other communication layers is shown in Figure G-24. A user of the Physical Layer exchanges the Data Link Layer PDU with its peer at another node by sending and receiving the Data Link PDU via the Physical Layer.

TABLE G-11. Example construction of data link transmission header.

Field Name	**Length**	**Value (Dec)**	**Value (Binary)** 2^n 2^0	**Field Fragments**	**Octet Value (Binary)** 2^n 2^0	**Octet Number**
Flag	8	126	*0*1111110	*0*1111110	*0*1111110	0
FEC	1	0	1	xxxxxxx1		
TDC	1	0	1	xxxxxx1x		
Scramble	1	0	1	xxxxx1xx		
Topology Update Id	3	0	*0*00	xx*0*00xxx		
Transmit Queue	10	0	*0*000000000	00xxxxxx *0*0000000	00*0*00111 *0*0000000	1 2
FCS	32	471931248	*0*0011100001000010001100101110000	*0*0011100 00100001 00011001 01110000	*0*0011100 00100001 00011001 01110000	3 4 5 6
Flag	8	126	*0*1111110	*0*1111110	*0*1111110	7

1	**2**	**3**	**85**	**86**	**87**
0x64f2	0xf296	0x905e	0xe098	0x0000	0x0000
0110010011110010	1111001010010110	1001000001011110	1110000010011000	0000000000000000	0000000000000000

FIGURE G-24. Serial representation of physical layer transmission unit.

G.3.7.1 Physical Layer Processing Example. The Physical layer encodes data submitted by the data link layer in a format to meet the physical media's requirements. This example does not address the electrical or mechanical functions normally associated with the physical layer protocols. At the physical layer the transmission header is extracted and the TWC is calculated, the Transmission header is FEC & TDC encoded.(Note the other physical layer functions (COMSEC, DMTD, etc) are not shown in this example.

TWC	*Transmission Header*	*Data Link Frame*

G.3.7.1.1 Transmit Word Count (TWC). TWC is calculated across the data link frame plus the size of the encoded Transmission Header & TWC size (encoded Transmission Header & TWC [10.5 16 bit words]). Therefore this Physical layer PDU's TWC would be calculated as follows:

TWC = encoded data link frame + encoded Transmission Header and TWC
TWC = 72 words + 10.5 words (rounded up to nearest word)
TWC = 83 words

TWC (83)	*Transmission Header*	*Data Link Frame*

Transmission header including TWC (size in bits: 0x004C)
0xca07 0xee00 0x01c2 0x1197 0x07e0

G.3.7.1.2 FEC & TDC of Transmission Header. The Transmission Header must have FEC & TDC encoding applied. Below is the Transmission Header in the different stages of FEC & TDC:

a. Transmission header/with TWC after FEC (Golay 24,12) (size in bits: 0x00a8)
Golay (24,12) is derived form Golay (23,12):See paragraph F 4.1 for details.
0xca0f 0x587e 0xe806 0x0000 0x001c 0x20c8 0x1191 0xfe70 0x75a4 0xe005 0x2600

b. Transmission header/with TWC after TDC (7,24)(size in bits 0x00a8)
0x838d 0x1aed 0x0a30 0x0448 0x8950 0x6c10 0xe047 0x1d30 0x3c49 0x89d2 0x8000

G.3.7.1.3 The Physical Layer PDU. Complete message including 64-bit frame synchronization, TWC, transmission header, and data link frame. (size in bits 0x0568):

0x64f2 0xf296 0x905e 0xadd9 0x838d 0x1aed 0x0a30 0x0448
0x8950 0x6c10 0xe047 0x1d30 0x3c49 0x89d2 0x8086 0x2308
0x882f 0x7f18 0xc1ee 0x2e91 0x58be 0x2084 0x47a5 0x9c47
0x9f64 0x0356 0x01e8 0x0533 0xf1ac 0xe00d 0x106d 0x958e
0x880f 0x50ca 0x80d4 0xa322 0x85b2 0xe000 0x009c 0x389e
0x09c8 0x615a 0x199c 0x580e 0x7b3c 0xfaa5 0x39b4 0xb8cd
0x81a2 0xf2b2 0x6833 0x8116 0x70c4 0x6bb3 0x283f 0x9157
0x1225 0xeaa5 0x78e8 0x2b84 0x29ce 0xcb00 0x00db 0x40cd
0xa0fa 0xc0d4 0x4000 0x005d 0x401a 0x00d4 0xe0ce 0x4043
0xc0c3 0x80cf 0x40fd 0x80b1 0x606e 0x00aa 0xe0d1 0xc0ee
0x60e0 0x401f 0xa07c 0xe08f 0xe098 0x0000 0x0000

APPENDIX H

INTRANET TOPOLOGY UPDATE

H.1. General.

H.1.1. Scope. This appendix describes a procedure for active intranet topology updates. The intranet is defined as all processors and CNRs within a single transmission channel.

H.1.2. Application. This appendix is a mandatory part of MIL-STD-188-220. The information contained herein is intended for compliance.

H.2. Applicable Documents. This section is not applicable to this appendix.

H.3. Problem Overview. Figure H-1 shows a sample extended CNR network. Each node labeled A through H is considered to be a radio with an associated communication processor. The dotted ovals indicate subsets of connectivity. Figure H-2 is a link diagram of the sample network. Assuming the nodes know nothing about neighbor nodes that are more than 1 hop away, they need to exchange connectivity information. The topology update packet is used to exchange topology information to build up a more complete view of the intranet's topology at every node.

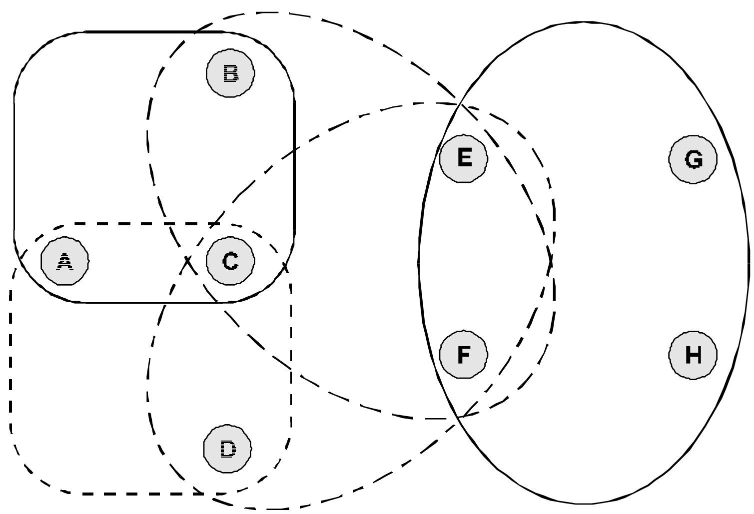


FIGURE H-1. Sample intranet.

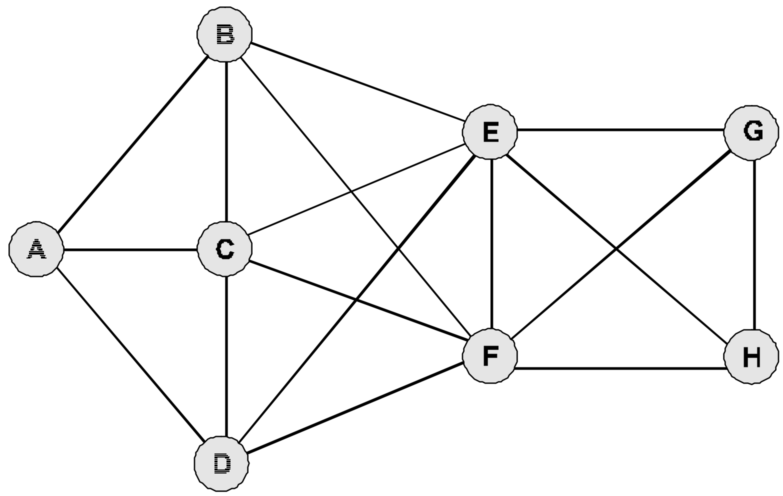


FIGURE H-2. Link diagram of sample network.

H.3.1. Routing Trees. Each node should store topology information as a routing tree graph. Considering the network in Figure H-2, Figure H-3 shows the routing tree for nodes A and C prior to the exchange of any topology information. The routing trees for A and C contain only their nearest neighbors - those nodes which they can talk to directly. Similar graphs would exist for all other nodes..

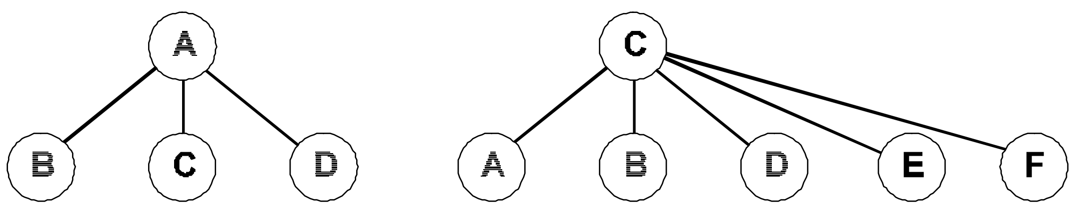


FIGURE H-3. Routing tree for nodes A and C.

H.4. Topology Updates

H.4.1. Exchanging Routing Trees. Nodes in the network gain more topology information by multicasting their individual routing trees to their nearest neighbor nodes. This exchange of routing trees will percolate more complete topology information through the network. For example, assume the routing trees for all nodes in Figure H-2 initially contain only nearest neighbors (nodes who are in direct communication with the given node). If node C multicasts its topology information to all nodes one hop away (those which are nearest neighbors), all neighbor nodes integrate C's routing tree into their own. Node A would integrate the graph for Node C into its routing tree as shown in Figure H-4.

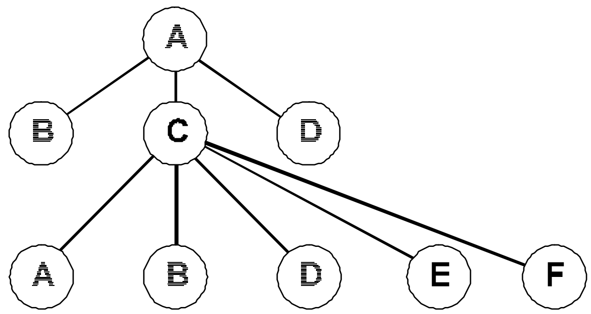


FIGURE H-4. Concatenated routing tree for node A.

Before the routing tree is saved, Node A prunes any successive instances of itself. For instance, in Figure H-4, the link from A to C is the same as the link from C to A; therefore, the link from C to A is removed. All redundant identical links are also pruned. These are links with the order of the end points reversed.

H.4.2. Topology Tables. The topology table for A is shown in Table H-1. It assumes no nodes are in quiet mode, all nodes can participate in relay, and all links have a cost of 1. The actual link layer addresses for the nodes would be placed into the table in place of the symbols A, B, C, etc. The extension bit in the address octet would always be set to 0 for topology updates.

TABLE H-1. Topology table for node A.

Node Address	Node Predecessor	Hops	Cost	NR	Quiet
B	A	1	1	0	0
C	A	1	1	0	0
D	A	1	1	0	0
B	C	2	1	0	0
D	C	2	1	0	0
E	C	2	1	0	0
F	C	2	1	0	0

There are two entries for node B indicating that there are two paths from A to B. This table could be immediately copied to the respective fields of a topology update packet. The predecessor address is not included in the topology update packet for nearest neighbor nodes because the predecessor is, by definition, the originator node.

H.4.3. Sparse Routing Trees. Exchanging full routing tree tables provides full topology information; however, the amount of data in the routing tree gets very large, especially for fully connected nets. The number of links in a fully connected net with n nodes is n(n-1)/2. Although full routing trees should be stored by a node, exchanging these routing trees may consume too much bandwidth. A smaller copy of the full routing tree (called a sparse routing tree) should be prepared for transmission to neighbor nodes. To reduce the number of branches in the routing tree, some of the paths to duplicate nodes on the tree are pruned according to following rules:

a. Only the shortest paths from the root node to another node are retained.

b. For redundant paths from a root node to a another node which are the same length (same number of links in the routing tree), at most 2 are retained. Some redundancy in paths is necessary for volatile networks.

For the previous example, the path from C to B and C to D would be pruned, since there are already shorter paths from A to C and A to D. The pruning yields the sparse routing tree in Figure H-5 and Table H-2.

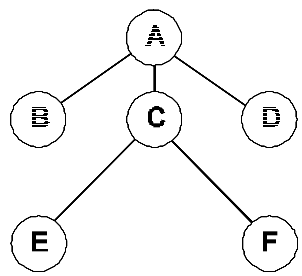


FIGURE H-5. Sparse routing tree for node A.

TABLE H-2. Sparse routing tree for node A.

Node Address	Node Predecessor	Hops	Cost	NR	Quiet
B	A	1	1	0	0
C	A	1	1	0	0
D	A	1	1	0	0
E	C	2	1	0	0
F	C	2	1	0	0

The final routing tree for Node A, after all the nodes exchange their sparse routing trees, is shown in Figure H-6 and Table H-3. Note that figure H-6 shows more than 2 paths between nodes G and A and H and A; however, the sparse routing tree table, which is the information actually transmitted, shows only two entries for nodes G and H. The pruning rules stated above have not been violated. They have been applied to the entries in the sparse routing table. The sparse routing graph is deduced from the table. Thus, quite a few redundant paths can be derived from the structure of the sparse routing table.

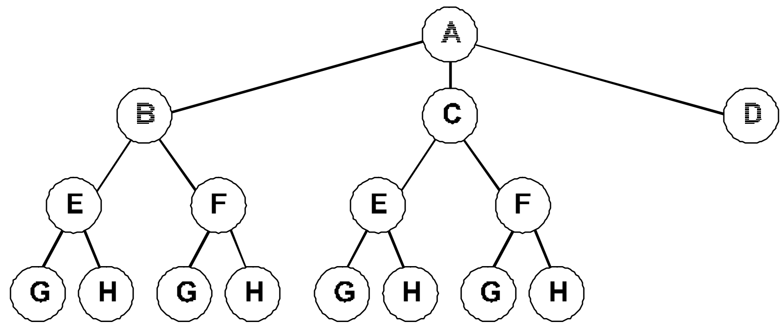


FIGURE H-6. Final routing tree for node A.

TABLE H-3. Final routing tree for node A.

Node Address	Node Predecessor	Hops	Cost	NR	Quiet
B	A	1	1	0	0
C	A	1	1	0	0
D	A	1	1	0	0
E	B	2	1	0	0
F	B	2	1	0	0
E	C	2	1	0	0
F	C	2	1	0	0
G	E	3	1	0	0
H	E	3	1	0	0
G	F	3	1	0	0
H	F	3	1	0	0

H.4.4. Rules For Exchanging Topology Updates. Topology update packets are transmitted exclusively using a global multicast address.

H.4.4.1. Topology update triggers. Topology updates are triggered for node I by the following:

a. Node I detects a failed link and the link is to a node that is not a static node (link quality =7)

b. Node I detects a new or recovered link and the link is to a node that is not a static node (link quality =7)

c. Node I detects a change in the quality of a link - applicable only if link costs are used.

d. Node I receives a topology update from another node which modifies its sparse routing tree.

e. Node I changes its Quiet Mode status and wishes to announce this change.

f. Node I changes its relay capability status.

g. Node I receives a topology update request.

H.4.4.2. Sending topology update messages. Optimally, topology updates should be concatenated with other traffic for queuing by the link layer. Topology Update Messages are sent to the global multicast address using Type 1 Unnumbered Information Frames which are not acknowledged. The precedence of the Topology Update Message is user configurable.

The updates should be transmitted no more often than once every MIN_UPDATE_PER. MIN_UPDATE_PER is measured in minutes and is set by the network administrator when the nodes are configured. The network administrator can disable topology update transmission by setting MIN_UPDATE_PER to zero. Update packets are superseded by newer packets if they have not been queued at the link layer.

H.4.5. Non-relayers. In the Topology Update broadcast by non-relayers, the non-relayer indicates its status by setting the NR bit to one in its entry of the Topology Update message. Additionally, the non-relayer includes all one-hop, and only one-hop, neighbors (because relaying by this node is not permitted). Non-relayer nodes remain in the sparse routing trees; however, they must not have any subsequent branches. Their entries in the routing table must have the NR bit set to 1.

H.4.6. Quiet Nodes. Nodes in the quiet state may appear in the sparse routing tables and in update packets with the QUIET bit set to 1; however, they must not have any subsequent branches in the routing tree. Nodes wishing to announce that they are entering quiet mode must add a separate entry into the sparse routing table and update packets with NODE ADDRESS and NODE PREDECESSOR set to their own address and the QUIET bit set to 1.

H.4.7. Topology Update Request Messages. The Topology Update Request Message is triggered whenever there is a mismatch between the topology update ID received from a station and the value that had been stored previously. The Topology Update Request message may also be sent whenever a data link transmission is detected from a previously unknown neighbor. The Topology Update Request message uses a Type 1 Unnumbered Information frame which is not acknowledged and is addressed according to paragraphs 5.4.1.1.7, 5.4.1.1.9, and 5.4.1.3. The Topology Update Request message is addressed to specific stations at the Intranet layer and may be sent to the global multicast address at the data link layer. The precedence of the Topology Update Request Message is user-configurable. The Topology Update Request Message may be sent no more often than MIN_UPDATE_PER/2. This constant allows up to two requests to be sent to a node while the node is waiting for the MIN_UPDATE_PER timer to expire.

APPENDIX I

SOURCE DIRECTED RELAY

I.1. General.

I.1.1 Scope. This appendix describes a procedure for relaying packets across a CNR intranet using source directed routes. The intranet is defined as all processors and CNRs within a single transmission channel.

I.1.2 Application. This appendix is a mandatory part of MIL-STD-188-220. The information contained herein is intended for compliance.

I.2. Applicable Documents. None.

I.3. Problem Overview. Intranet relaying is required when nodes in a intranet need to communicate, but are not nearest neighbors capable of hearing one another's radio transmissions.

I.4. Procedure.

I.4.1 Forward Routing. Source Directed Relay provides a simple non-dynamic procedure for relaying a packet from an originator to one or more destinations. The source must calculate the path through the intranet network to reach each destination. These paths are based on the topology and connectivity table. The specific source directed route for each destination must be encoded into the intranet header. If the routes for two or more destinations share common links along the paths, the two paths should be merged together. As a result of this, the resulting paths should not have any common nodes.

The address of successive relayers, destinations, and their associated status bytes are placed in the intranet header in order of progressing through the routing tree. Nodes which are one hop away and destinations only are placed into the Intranet Header first with their DES bit set to 1. The next entries into the Intranet Header are the relay paths which may include nodes which are relayers and destinations. Each relay path starting at the source is completed before another relay path with its origin at the source is begun. Within the status byte for each relayer the REL bit is set to 1 and S/D is set to 0. If the relayer is also a destination in addition to being a relayer, the DES bit is set to 1. If there are multiple destinations that are not relayers following a relayer, each of these destination addresses and their status bytes should be listed in the header after the relay node sequentially in the order of their appearance in the path. Within this group the extension bit within the destination/relay address field is not used. The last address can be determined from the Intranet header length. The last address in a group can be determined from the DIS field of the Destination/Relay Status Byte defined in 5.4.1.1.7.

All destinations in the relay path that are required to provide end-to-end intranet acknowledgments have set the ACK bit in their status bytes to 1. For all destinations, the **DISTANCE** field is set to the number of hops between the originator and the ultimate destination host for the relay.

I.4.2 End-to-end Acknowledgments. End-to-end Acknowledgments are formed by the *i*th final destination nodes upon receipt of an intranet header with **ACK** bit set in **DESTINATION STATUS BYTE** for the *i*th destination. The **MESSAGE ID** for the packet to be acknowledged is retained. The message type is set to 1. The path between the originator node and the *i*th destination is reversed. All intermediate destinations are removed. The path will contain one originator, one destination, and the relayers. The **DES** bit in the status bytes for all relayers is set to 0, indicating they perform relay only. No data is carried with an end-to-end acknowledgment packet; just the intranet header.

I.5. Examples. To illustrate Source Directed Relay procedures consider the sample network link diagram in Figure I-1 and final routing tree in Figure I-2. Table I-1 gives specific addresses for the nodes labeled A, B, C, D, E, F, G and H. To maintain consistency with other sections of MIL-STD-188-220, the least significant bit (LSB) is presented to the left of the figures in this appendix.

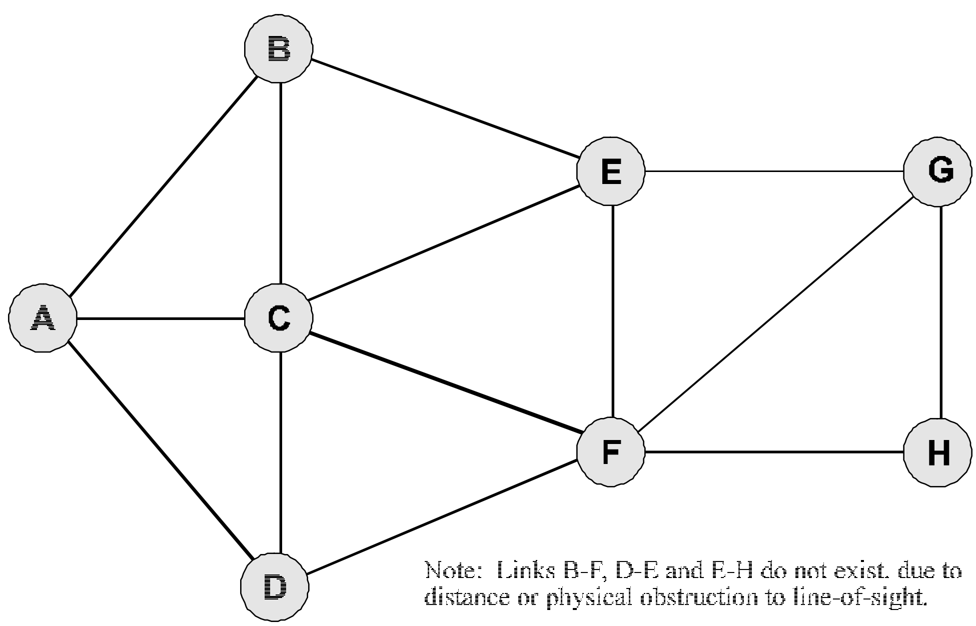


FIGURE I-1. Link diagram of a sample network.

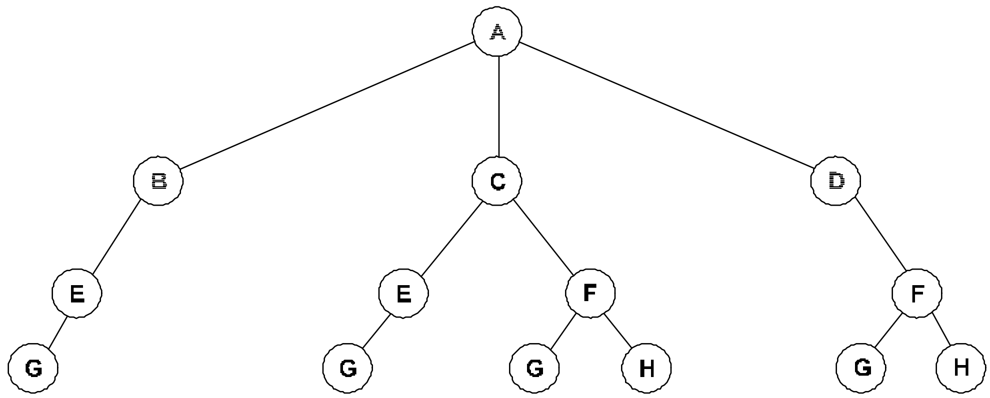


FIGURE I-2. Final routing tree for node A.

TABLE I-1. Sample Node Addresses

Node	LSB						MSB	Address
A	x	1	1	1	1	0	0	0	15
B	x	0	0	1	0	0	0	0	4
C	x	1	0	1	0	0	0	0	5
D	x	0	1	1	0	0	0	0	6
E	x	1	1	1	0	0	0	0	7
F	x	0	0	0	1	0	0	0	8
G	x	1	0	0	1	0	0	0	9
H	x	0	1	0	1	0	0	0	10

I.5.1. EXAMPLE 1. Assume that node A has a packet bound for node G alone. Node A's Sparse Routing Tree provides the following potential paths to Node G: A-B-E-G, A-C-E-G, A-C-F-G and A-D-F-G. Assuming that all paths have the same quality and cost, any path may be selected by Node A. In this example, path A-B-E-G is selected.

The following values are assigned to the Intranet Header in example 1:

MESSAGE TYPE = 4 (IP Packet)
TYPE_OF_SERVICE = 0000 0000
MESSAGE ID = 1
MAX_HOP_COUNT = 3 (Distance from node A to node G)
ORIGINATOR ADDRESS = 15 (node A)
STATUS BYTE 1 = 10010000 (DIS=1, REL=Yes, DES=No, ACK=No)
DESTINATION 1 = 4 (node B)
STATUS BYTE 2 = 01010000 (DIS=2, REL=Yes, DES=No, ACK=No)
DESTINATION 2 = 7 (node E)

STATUS BYTE 3 = 11000010 (DIS=3, REL=No, DES=Yes, ACK=No)
DESTINATION 3 = 9 (node G)
HEADER LENGTH = 12 octets

Figure I-3 shows the complete Intranet Header for example 1. Note that the LSB in all destination addresses is 0 except for the last destination address (node G).

LSB	*1*	*2*	*3*	*4*	*5*	*6*	*7* *MSB*
VERSION NUMBER				MESSAGE TYPE			
0	0	0	0	0	0	1	0
INTRANET HEADER LENGTH							
0	0	1	1	0	0	0	0
TYPE OF SERVICE							
0	0	0	0	0	0	0	0
MESSAGE IDENTIFICATION NUMBER							
1	0	0	0	0	0	0	0
MAX HOP COUNT				SPARE			
1	1	0	0	0	0	0	0
ORIGINATOR ADDRESS							
0	1	1	1	1	0	0	0
DESTINATION/RELAY STATUS BYTE 1							
1	0	0	1	0	0	0	0
DESTINATION/RELAY ADDRESS 1							
0	0	0	1	0	0	0	0
DESTINATION/RELAY STATUS BYTE 2							
0	1	0	1	0	0	0	0
DESTINATION/RELAY ADDRESS 2							
0	1	1	1	0	0	0	0
DESTINATION/RELAY STATUS BYTE 3							
1	1	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 3							
1	1	0	0	1	0	0	0

FIGURE I-3. Example 1 intranet header.

I.5.2. EXAMPLE 2. Assume that node A has a packet bound for nodes G and H. Node A's Sparse Routing Tree provides the following potential paths to nodes G and H: A-B-E-G, A-C-E-G, A-C-F-G, A-C-F-H, A-D-F-G, and A-D-F-H. Of these potential paths, the most economical choices are those that use node F for relaying: A-C-F-G, A-D-F-G, A-C-F-H, and A-D-F-H. Although paths A-B-E-G and A-C-E-G are viable paths to node G, they would unnecessarily increase processing at nodes B and E, and would increase the size of the Intranet Header in this example. In this example the selected paths are A-C-F-G and A-C-F-H.

The following values are assigned to the Intranet Header in example 2:

MESSAGE TYPE = 4 (IP Packet)
TYPE_OF_SERVICE = 0000 0000

MESSAGE ID = 2
MAX_HOP_COUNT = 3 (Distance from node A to nodes G and H)
ORIGINATOR ADDRESS = 15 (node A)
STATUS BYTE 1 = 10010000 (DIS=1, REL=Yes, DES=No, ACK=No)
DESTINATION 1 = 4 (node C)
STATUS BYTE 2 = 01010000 (DIS=2, REL=Yes, DES=No, ACK=No)
DESTINATION 2 = 8 (node F)
STATUS BYTE 3 = 11000010 (DIS=3, REL=No, DES=Yes, ACK=No)
DESTINATION 3 = 9 (node G)
STATUS BYTE 4 = 11000010 (DIS=3, REL=No, DES=Yes, ACK=No)
DESTINATION 4 = 10 (node H)
HEADER LENGTH = 14 octets

Figure I-4 shows the complete Intranet Header for example 2. Note that the LSB in all destination addresses is 0 except for the last destination address (node H).

0 *LSB*	*1*	*2*	*3*	*4*	*5*	*6*	*7* *MSB*
VERSION NUMBER				MESSAGE TYPE			
0	0	0	0	0	0	1	0
INTRANET HEADER LENGTH							
0	1	1	1	0	0	0	0
TYPE OF SERVICE							
0	0	0	0	0	0	0	0
MESSAGE IDENTIFICATION NUMBER							
0	1	0	0	0	0	0	0
MAX HOP COUNT				SPARE			
1	1	0	0	0	0	0	0
ORIGINATOR ADDRESS							
0	1	1	1	1	0	0	0
DESTINATION/RELAY STATUS BYTE 1							
1	0	0	1	0	0	0	0
DESTINATION/RELAY ADDRESS 1							
0	0	0	1	0	0	0	0
DESTINATION/RELAY STATUS BYTE 2							
0	1	0	1	0	0	0	0
DESTINATION/RELAY ADDRESS 2							
0	0	0	0	1	0	0	0
DESTINATION/RELAY STATUS BYTE 3							
1	1	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 3							
0	1	0	0	1	0	0	0
DESTINATION/RELAY STATUS BYTE 4							
1	1	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 4							
1	0	1	0	1	0	0	0

FIGURE I-4. Example 2 intranet header.

I.5.3. EXAMPLE 3. In the third example, node A wishes to deliver a packet to nodes D, E, F, G and H. In this case node A again would select the most economical path to each destination, taking into consideration the impacts on network traffic and Intranet header size. Table I-2 lists the potential and selected paths from node A to each of the intended destinations.

A similar process would be used to select economical paths to relay nodes, such as node C. The shortest path to the most distant nodes G and H are reviewed to determine whether the relay nodes C and F are also destinations. Note that node F is both a destination and a relay while node C is a relay node only.

Destination Node	Potential Paths	Selected Path
D	A-D	A-D
E	A-B-E A-C-E	A-C-E
F	A-C-F A-D-F	A-C-F
G	A-B-E-G A-C-E-G A-C-F-G A-D-F-G	A-C-F-G
H	A-C-F-H A-D-F-H	A-C-F-H

TABLE I-2. Paths used in Example 3.

The following values are assigned to the Intranet Header in example 3:

MESSAGE TYPE = 4 (IP Packet)
TYPE_OF_SERVICE = 0000 0000
MESSAGE ID = 3
MAX_HOP_COUNT = 3 (Distance from node A to nodes G and H)
ORIGINATOR ADDRESS = 15 (node A)
STATUS BYTE 1 = 10000010 (DIS=1, REL=No, DES=Yes, ACK=No)
DESTINATION 1 = 6 (node D)
STATUS BYTE 2 = 10010000 (DIS=1, REL=Yes, DES=No, ACK=No)
DESTINATION 2 = 5 (node C)
STATUS BYTE 3 = 01000010 (DIS=2, REL=No, DES=Yes, ACK=No)
DESTINATION 3 = 7 (node E)
STATUS BYTE 4 = 01010010 (DIS=2, REL=Yes, DES=Yes, ACK=No)
DESTINATION 4 = 8 (node F)
STATUS BYTE 5 = 11000010 (DIS=3, REL=No, DES=Yes, ACK=No)
DESTINATION 5 = 9 (node G)

STATUS BYTE 6 = 11000010 (DIS=3, REL=No, DES=Yes, ACK=No)
DESTINATION 6 = 10 (node H)
HEADER LENGTH = 18 octets

Figure I-5 shows the complete Intranet Header for example 3. Note that the LSB in all destination addresses is 0 except for the last destination address (node H).

0 LSB	*1*	*2*	*3*	*4*	*5*	*6*	*7 MSB*
VERSION NUMBER				MESSAGE TYPE			
0	0	0	0	0	0	1	0
INTRANET HEADER LENGTH							
0	1	0	0	1	0	0	0
TYPE OF SERVICE							
0	0	0	0	0	0	0	0
MESSAGE IDENTIFICATION NUMBER							
1	1	0	0	0	0	0	0
MAX HOP COUNT				SPARE			
1	1	0	0	0	0	0	0
ORIGINATOR ADDRESS							
0	1	1	1	1	0	0	0
DESTINATION/RELAY STATUS BYTE 1							
1	0	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 1							
0	0	1	1	0	0	0	0
DESTINATION/RELAY STATUS BYTE 2							
1	0	0	1	0	0	0	0
DESTINATION/RELAY ADDRESS 2							
0	1	0	1	0	0	0	0
DESTINATION/RELAY STATUS BYTE 3							
0	1	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 3							
0	1	1	1	0	0	0	0
DESTINATION/RELAY STATUS BYTE 4							
0	1	0	1	0	0	1	0
DESTINATION/RELAY ADDRESS 4							
0	0	0	0	1	0	0	0
DESTINATION/RELAY STATUS BYTE 5							
1	1	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 5							
0	1	0	0	1	0	0	0
DESTINATION/RELAY STATUS BYTE 6							
1	1	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 6							
1	0	1	0	1	0	0	0

FIGURE I-5. Example 3 intranet header created by node A (originator)

I.5.4. RELAY PROCESSING. Although the separate examples 1,2,3 all have diverse paths, they would all require the same number data link information frames for delivery (one). The UI, I, or DIA frame would be transmitted to each destination simultaneously. Addressed destinations would perform the required data link layer processing described in 5.3 and pass the information field of the frame to the Intranet layer for further processing.

The Intranet header is scanned for the node's data link layer address. When found, the previous octet - the Destination/Relay Status byte - is inspected. If the Relay bit is not set and the destination bit is set, the data portion following the Intranet header is passed to the next higher protocol layer for further processing. If the Relay bit is set, Relay processing is required. If both the Relay bit and the Destination bit are set, Relay processing is performed before the passing data portion of the frame to the next higher protocol layer for further processing. Relay processing involves the following steps:

a. Scan forward until the relayer node sees its own address.

b. Scan toward the end of the header looking for all nodes whose DES bit is set and whose distance is one hop greater than your own. Terminate the scan when a distance less than or equal to your own or the end of the header is found. Save the addresses.

c. While scanning forward until a hop distance less than or equal to your own is found, find all relay addresses that are one hop away from your address and save these addresses.

d. Remove all duplicate saved addresses and pass the remaining addresses to the data link layer to form a multi-addressed information frame containing the Intranet header and data.

The following sections discuss the relay processing at each of the downstream relayers in Example 3. There are two options when filling out the Intranet Header Address Field at the relay nodes. The relay nodes may copy the Address Field and place it into the relay packet intact or they may delete the addresses which have no impact on forwarding or return of a network layer acknowledgment. If the implementor chooses to leave the address field intact, the address field in Figure I-5 is used at every relayer. If the implementor chooses to compress the address field to save transmitted bytes, the following paragraphs dictate the method for compression. There is no interoperability problem regardless of which of these two methods are implemented.

I.5.4.1 RELAY PROCESSING AT NODE C. Node C is a relay node, but not a destination node. Node C is responsible for relaying the information to nodes E F, G and H. Node C assigns the following values to the Intranet Header in example 3:

MESSAGE TYPE = 4 (IP Packet)
TYPE_OF_SERVICE = 0000 0000
MESSAGE ID = 3
MAX_HOP_COUNT = 2 (Original MAX_HOP_COUNT - 1)
ORIGINATOR ADDRESS = 15 (node A)

STATUS BYTE 1 = 10010000 (DIS=1, REL=Yes, Des=No, ACK=No)
DESTINATION 1 = 5 (node C)
STATUS BYTE 2 = 01000010 (DIS=2, REL=No, DES=Yes, ACK=No)
DESTINATION 2 = 7 (node E)
STATUS BYTE 3 = 01010010 (DIS=2, REL=Yes, DES=Yes, ACK=No)
DESTINATION 3 = 8 (node F)
STATUS BYTE 4 = 11000010 (DIS=3, REL=No, DES=Yes, ACK=No)
DESTINATION 4 = 9 (node G)
STATUS BYTE 5 = 11000010 (DIS=3, REL=No, DES=Yes, ACK=No)
DESTINATION 5 = 10 (node H)
HEADER LENGTH = 16 octets

Figure I-6 shows the complete Intranet Header created by Node C.

I.5.4.2. RELAY PROCESSING AT NODE F. Node F is both a destination and a relayer with relay responsibilities to nodes G and H. Node F assigns the following values to the Intranet Header in example 3:

MESSAGE TYPE = 4 (IP Packet)
TYPE_OF_SERVICE = 0000 0000
MESSAGE ID = 3
MAX_HOP_COUNT = 1 (Received MAX_HOP_COUNT - 1)
ORIGINATOR ADDRESS = 15 (node A)
STATUS BYTE 1 = 10010000 (DIS=1, REL=Yes, DES=No, ACK=No)
DESTINATION 1 = 5 (node C)
STATUS BYTE 2 = 01010010 (DIS=2, REL=Yes, DES=Yes, ACK=No)
STATUS BYTE 2 = 8 (node F)
STATUS BYTE 3 = 11000010 (DIS=3, REL=No, DES=Yes, ACK=No)
DESTINATION 3 = 9 (node G)
STATUS BYTE 4 = 11000010 (DIS=3, REL=No, DES=Yes, ACK=No)
DESTINATION 4 = 10 (node H)
HEADER LENGTH = 14 octets

Figure I-7 shows the complete Intranet Header created by Node F.

LSB	*1*	*2*	*3*	*4*	*5*	*6*	*7* *MSB*
VERSION NUMBER				MESSAGE TYPE			
0	0	0	0	0	0	1	0
INTRANET HEADER LENGTH							
0	0	0	0	1	0	0	0
TYPE OF SERVICE							
0	0	0	0	0	0	0	0
MESSAGE IDENTIFICATION NUMBER							
1	1	0	0	0	0	0	0
MAX HOP COUNT				SPARE			
0	1	0	0	0	0	0	0
ORIGINATOR ADDRESS							
0	1	1	1	1	0	0	0
DESTINATION/RELAY STATUS BYTE 1							
1	0	0	1	0	0	0	0
DESTINATION/RELAY ADDRESS 1							
0	1	0	1	0	0	0	0
DESTINATION/RELAY STATUS BYTE 2							
0	1	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 2							
0	1	1	1	0	0	0	0
DESTINATION/RELAY STATUS BYTE 3							
0	1	0	1	0	0	1	0
DESTINATION/RELAY ADDRESS 3							
0	0	0	0	1	0	0	0
DESTINATION/RELAY STATUS BYTE 4							
1	1	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 4							
0	1	0	0	1	0	0	0
DESTINATION/RELAY STATUS BYTE 5							
1	1	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 5							
1	0	1	0	1	0	0	0

FIGURE I-6. Example 3 intranet header for node C (relay node)

0 LSB	*1*	*2*	*3*	*4*	*5*	*6*	*7 MSB*
VERSION NUMBER				MESSAGE TYPE			
0	0	0	0	0	0	1	0
INTRANET HEADER LENGTH							
0	1	0	1	0	0	0	0
TYPE OF SERVICE							
0	0	0	0	0	0	0	0
MESSAGE IDENTIFICATION NUMBER							
1	1	0	0	0	0	0	0
MAX HOP COUNT				SPARE			
1	0	0	0	0	0	0	0
ORIGINATOR ADDRESS							
0	1	1	1	1	0	0	0
DESTINATION/RELAY STATUS BYTE 1							
1	0	0	1	0	0	0	0
DESTINATION/RELAY ADDRESS 1							
0	1	0	1	0	0	0	0
DESTINATION/RELAY STATUS BYTE 2							
0	1	0	1	0	0	1	0
DESTINATION/RELAY ADDRESS 2							
0	0	0	0	1	0	0	0
DESTINATION/RELAY STATUS BYTE 3							
1	1	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 3							
0	1	0	0	1	0	0	0
DESTINATION/RELAY STATUS BYTE 4							
1	1	0	0	0	0	1	0
DESTINATION/RELAY ADDRESS 4							
1	0	1	0	1	0	0	0

FIGURE I-7. Example 3 intranet header created by node F (relay and destination node)

APPENDIX J

ROBUST COMMUNICATIONS PROTOCOL

J.1. General.

J.1.1. Scope. This Appendix describes the interoperability and technical requirements for the robust communications protocol for DMTD and interfacing C4I systems (DTEs). This Appendix applies only to HAVEQUICK II compatible systems that require interoperability with radios that do not have data buffering or synchronization capability.

J.1.2 Application. This Appendix is a mandatory part of MIL-STD-188-220. The information contained herein is intended for compliance.

J.2. Applicable Documents. This section is not applicable to this Appendix.

J.3. Introduction**.** This physical layer protocol provides the additional processing to aid the transfer of secure and non-secure digital data when concatenated with the link processing of the MIL-STD-188-220 protocol. The additional processing of this protocol allows for a higher level protocol with an error correcting capability equal to rate 1/2 Golay to transfer a burst of data containing up to 67,200 data symbols with better than 90% probability of success in a single transmission, this being over an active HAVEQUICK II compatible link with a random bit error rate of 0.1 or less. The second goal of this physical protocol is for the required performance to be achieved entirely in software using current systems with modest processing capability.

J.3.1 Physical Protocol Components. Three individually selectable processes are used to meet the performance requirement. The first is the application of rate 1/3 convolutional coding to combat high random bit error rates. The second is a provision for data scrambling. Scrambling at the physical layer is implemented simply as the multiplication of the transmit data with a pseudo random bit pattern. The third is a packetizing scheme that allows for the re-transmission of the data that was lost due to an HAVEQUICK II compatible frequency hop. The re-transmission is performed, and data recovered within the data burst and the data interruption is transparent to the higher level protocol. This packetizing scheme has been dubbed the Multi-Dwell protocol because it was formulated to allow a message to be transmitted over multiple HAVEQUICK II compatible hop dwells.

J.3.2 Optional rate 1/3 convolutional coding. The transmitting convolutional encoder generates three output bits for each input information bit. Figure J-1 shows an example of the encoding process for a constraint length (K) of 3. The encoder consists of a shift register equal in length to the constraint length. The data to encode is shifted from left to right one bit at a time. After each shift, three output bits are generated using the G1, G2, and G3 polynomials. The three encoded output bits are generated in the G1, G2, G3 order. The G2 output shall be inverted to provide some data scrambling capability. The convolutional encoding shift register is initialized to a state of zero when a transmission is requested. The first output bits are generated when the shift register contains the first upper layer bit to transmit, followed by all zeros. Upon detection of the robust synchronization pattern, the Viterbi decoder is initialized to make use of the knowledge of the initial encoder shift register state.

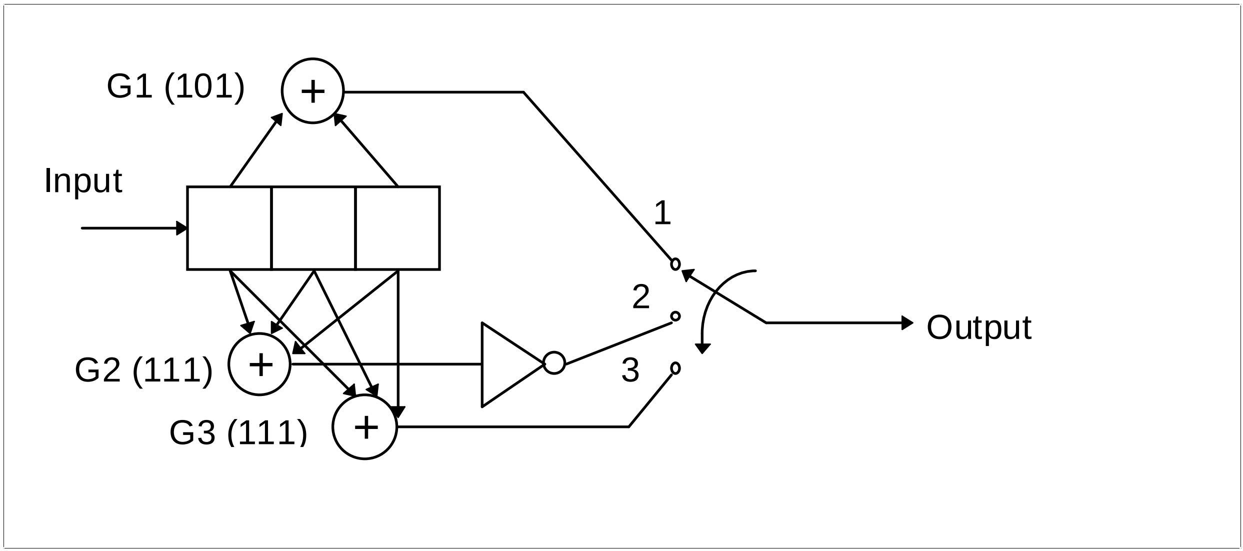


FIGURE J-1. Convolutional encoder with inverted G2 K=3.

Table J-1 lists the generator polynomials used for the three specified constraint lengths. The most significant bits of the octal representation of each polynomial are used for each polynomial.

TABLE J-1. Convolutional coding generator polynomials (octal).

Constraint Length	G1	G2	G3
3	5	7	7
5	52	66	76
7	554	624	764

Figure J-2 shows the relative error correcting capability of rate 1/3 convolutional coding in a random error environment using the Viterbi decoding algorithm with hard decisions. The performance was achieved using a trace back buffer length of 16, 32, and 64 for constraint lengths 3, 5, and 7 respectively. If the demodulator and decoder are components of the same subsystem, soft decision information from the demodulator can be used to further enhance the performance.

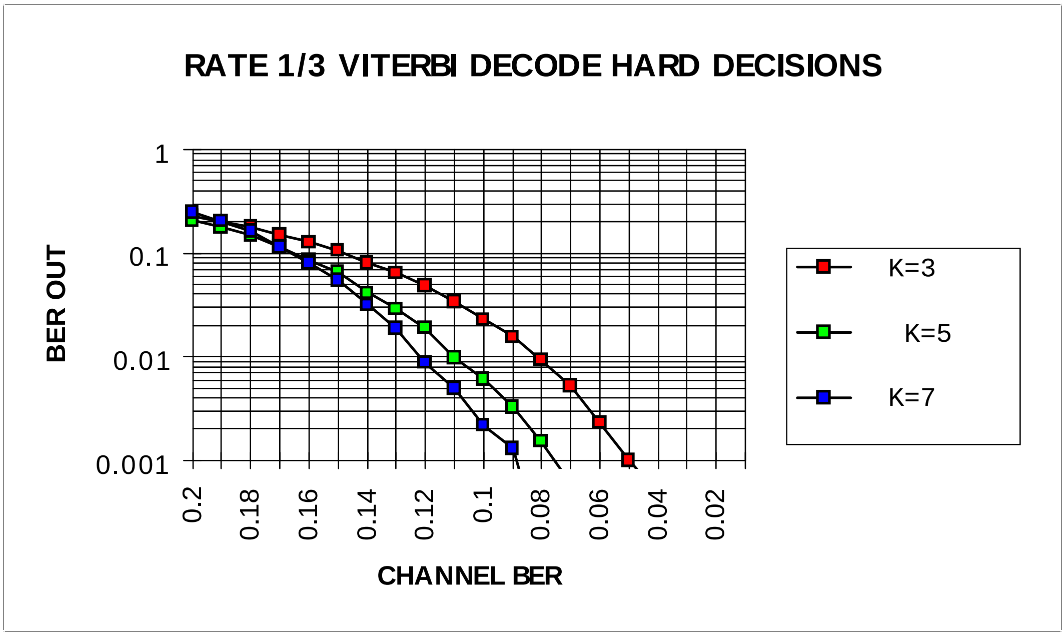


FIGURE J-2. Rate 1/3 convolutional coding performance for constraint lengths 3, 5, and 7.

J.3.3 Optional data scrambling. Physical layer data scrambling shall use the pseudo random bit generator specified in CCITT V.33 Annex A. The shift register shall be initialized to all zeros before the first bit of data is scrambled on transmission. On data reception, the descrambler shift register shall be initialized to zero before the first received data bit is descrambled. Figure J-3 shows the structure of the data scrambler and descrambler.

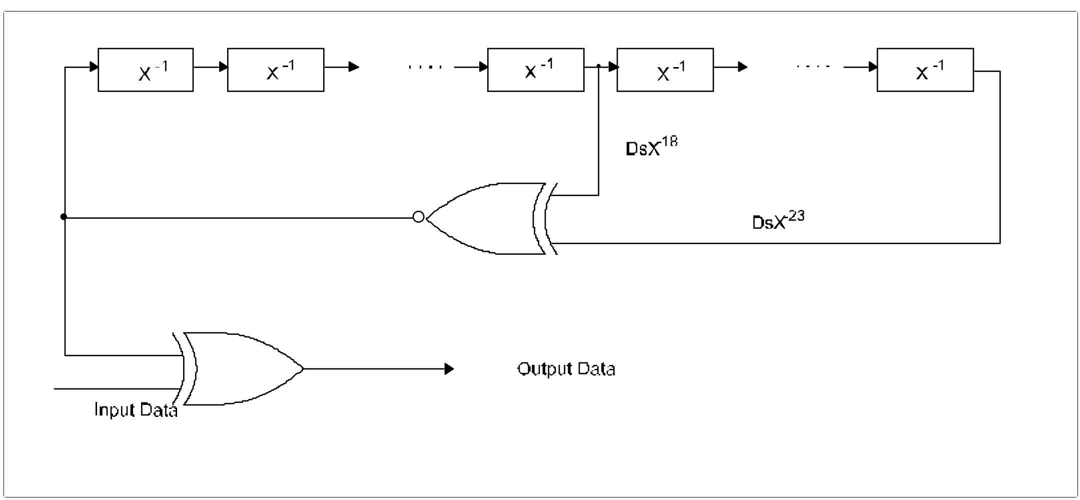


FIGURE J-3. Data scrambler structure.

J.3.4 Optional robust multi-dwell.

J.3.4.1 Multi-dwell packet format. When the HAVEQUICK II compatible radio is in active mode, multi-dwell packetizing shall be enabled. The multi-dwell packetizing described in this appendix assumes a physical level bit rate of 16 kbps. The format of each multi-dwell packet is shown in Figure J-4. Each packet consists of a start of packet (SOP) pattern and a segment counter followed by 6, 11 or 13 64-bit data segments.

J.3.4.2 Multi-dwell SOP field. The SOP pattern is a 32-bit (Figure J-5) or 64-bit (Figure J-6) pattern used for multi-dwell packet detection. The maximum number of bits in error should be set to match the bit error rate environment. For normal operation, it is recommended that the maximum number of bits in error be set to 13 for a 64-bit pattern, and to 3 for a 32-bit pattern. The length of the SOP pattern shall be determined by bits two three and four of the robust frame format.

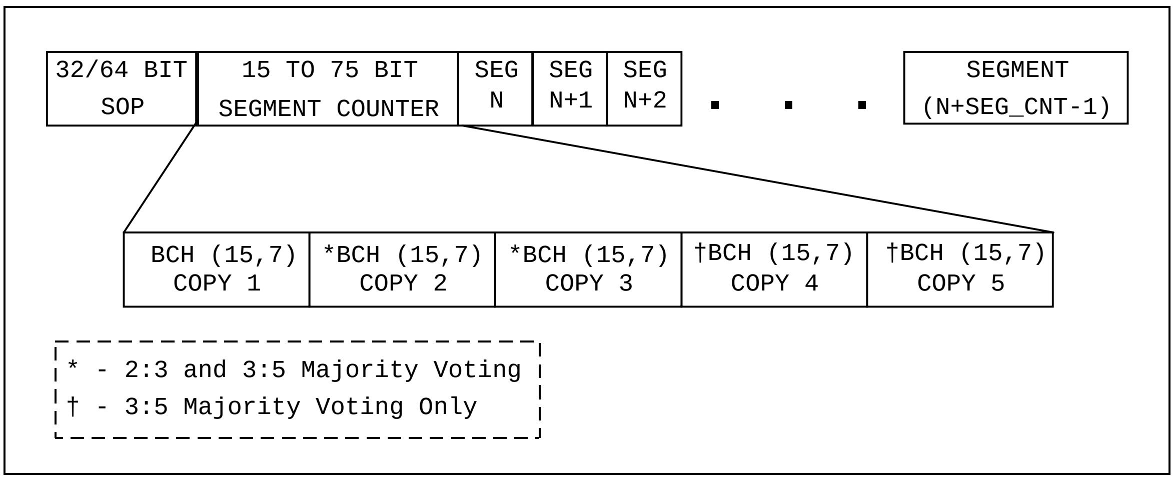


FIGURE J-4. Multi-dwell packet.

LSB	**MSB**
1010010011100010111100101000011010010000011111101010110011011101	

FIGURE J-5. Multi-dwell 64-bit SOP pattern.

LSB	MSB
00000011100100001001001110101110	

FIGURE J-6. Multi-dwell 32-bit SOP pattern.

J.3.4.3 Multi-dwell segment count field. The segment counter is a modulo 64 count of the first segment in the packet. The six required bits shall be encoded as 1, 3, or 5 BCH (15,7) codewords depending on bits 2, 3 and 4 of the robust frame format. The six-bit segment counter shall occupy the 6 least significant bits of the seven-bit BCH data field. The most significant bit of the data field shall be used as an end of frame flag which, when set, indicates that data transmission is complete. A multi-dwell packet marked with an end of frame flag shall contain only the SOP pattern and the segment count field used to make the segment number of the last non-fill data segment transmitted.

J.3.4.4 Multi-dwell data segments. Each multi-dwell packet shall contain 6, 11 or 13 consecutive 64-bit data segments. Unless a channel interruption is detected during the transmission of the packet, each data segment shall contain the next 64 bits supplied by the data link layer for transmission. The last multi-dwell packet shall contain pad bits and segments as necessary to complete the packet. The transmitted pad data shall be an alternating one, zero sequence.

J.3.4.5 Multi-dwell hop detection. The physical layer shall have the means of detecting or predicting communications link outages.

J.3.4.6 Multi-dwell transmit processing. Data received from the data link layer for transmission shall be broken into 64 bit segments for transmission. The data shall be packetized as stated in J.3.4.1. Packets shall be transmitted consecutively with the packet count subfield containing the count, modulo 64, of the first segment in the packet until a communications link outage is detected, at which time, the remainder of the data segments in the currently transmitted packet shall be filled with an alternating one/zero pattern. If the configurable hop recovery time (HRT), is greater than the time remaining to complete the transmission of the current packet, the alternating one/zero sequence shall be extended to the end of the HRT period. If a hop is detected during the multi-dwell packet synchronization field, or during the transmission of the first two segments, the entire packet shall be retransmitted. The first multi-dwell packet transmitted in a frame shall not contain the multi-dwell synchronization field. It is assumed that the segment count of the first packet is zero.

J.3.4.6.1 Hop data recovery time period. A configurable variable called the hop recovery time (HRT) shall be used to determine if the fill data transmitted following a hop must be extended to ensure that the following multi-dwell synchronization field can be received. The HRT is defined as the time period from the beginning of the transmitting radio frequency synthesizer frequency hop to the time that the bit synchronizer connected to the receiving radio can reliably demodulate data. Because different hop detection/ prediction methods flag the hop at different times relative to the beginning of the transmitting radio frequency synthesizer frequency slew, the configured HRT shall be internally adjusted to insure that different DTEs in a network can all use the same configurable HRT.

J.3.4.6.2 Data transmitted after a hop. The multi-dwell packet transmitted directly following a communications link outage shall retransmit data starting with the 64-bit segment preceding the segment that was being transmitted when the hop was detected.

J.3.4.6.3 Termination of transmission. After the final packet of the frame is transmitted, without a hop detected during a data segment containing actual data (not fill data), data transmission shall be terminated. To prevent receive delays caused by the receiver not being able to determine that the last data segment has been received, an optional truncated multi-dwell packet shall be sent with the end of frame flag set. The segment count associated with the end of frame flag shall mark the first non-fill data segment transmitted.

J.3.4.7 Multi-dwell receive processing. If the multi-dwell flag was set in the robust synchronization field, the receiver shall buffer the multi-dwell data packet. The segment count for the first multi-dwell packet in a frame shall be assumed to be 0. After the last packet bit is received, the receiver shall open the SOP correlator window. When the SOP pattern is recognized, the segment count is decoded using the combination of majority and BCH decoding specified in the robust synchronization field. After each new segment count is decoded, the buffered data for data segments lower in count than the new segment count are passed on to the next higher layer as received bits. The segments of the newly received packet are then buffered and held until it is verified that the buffered segments will not be re-transmitted.

J.3.4.7.1 Receive end of frame detection. The data remaining in the multi-dwell receive data buffer shall be provided to the higher level protocol when an end of frame condition is detected. The end of frame condition may be determined by the data demodulator, the optional multi-dwell end of frame flag, or by a message from the higher level protocol indicating that message reception is complete.

J.3.4.7.2 Optional soft decision information. When there is a very high link BER, a SOP pattern may not be recognized or the segment count may not be correctable. If fewer than three consecutive segment counts cannot be corrected the correct number of bits shall be supplied to the upper level protocol as to not cause a bit slip, and consequently, the loss of the remaining data in the frame. If convolutional coding is used with multi-dwell, it is suggested that soft decision information is supplied indicating the low quality of the received data resulting from a missed SOP pattern or an unrecoverable segment count.

J.3.4.8 Multi-dwell majority logic overhead choice. The choice of the amount of multi-dwell majority voting (MV) overhead is dependent on the expected link BER. Table J-2 gives an estimate of the maximum random BER supported for a 90% probability of passing a single frame of length 1536 bits, 7680 bits, and 67,200 bits with no errors introduced due to multi-dwell processing.

TABLE J-2. Maximum supported BER.

Segment Count MV	1536	7680	67,200
1 out of 1	0.055	0.03	0.016
2 out of 3	0.14	0.11	0.07
3 out of 5	0.2	0.14	0.12

J.3.4.9 Multi-dwell overhead. The multi-dwell protocol introduces an overhead that shall be considered in the network timing calculations. The overhead is a function of the radio hop rate, the multi-dwell segment count majority voting choice, and the message length. Table J-3 gives the equation to calculate the actual worst case realized data rate for each hop rate and majority logic combination. The numbers in table J-3 were run with a hop recovery time of 15.625 ms, a maximum radio timing drift over a 1/2 hour period, an instantaneous data rate of 16000 bits/second. The actual efficiency will depend upon the exact implementation, therefore the numbers in Table J-3 should be used as a guide only. The six-segment multi-dwell packet shall be used for protocol acknowledgments and other single TDC block messages. The calculated realized data rate shall be used for the bit rate of all data encapsulated by the multi-dwell protocol.

TABLE J-3. Multi-dwell overhead.

HOP	Multi-dwell overhead calculation			
RATE	MV 1:1, 11 segments	MV 2:3, 13 segments	MV 3:5, 13 segments	MV 3:5, 6 segments
0	$R/((0.3*10^{(-L*.00003)})+1.06)$	$R/((0.3*10^{(-L*.00003)})+1.16)$	$R/((0.2*10^{(-L*.00003)})+1.17)$	$R/((0.1*10^{(-L*.00003)})+1.36)$
1	$R/((0.6*10^{(-L*.00003)})+1.10)$	$R/((0.6*10^{(-L*.00003)})+1.21)$	$R/((.55*10^{(-L*.00003)})+1.23)$	$R/((0.3*10^{(-L*.00003)})+1.40)$
2	$R/((0.5*10^{(-L*.00003)})+1.15)$	$R/((0.5*10^{(-L*.00003)})+1.27)$	$R/((0.7*10^{(-L*.00003)})+1.30)$	$R/((0.4*10^{(-L*.00005)})+1.48)$
3	$R/((0.5*10^{(-L*.00003)})+1.20)$	$R/((0.4*10^{(-L*.00002)})+1.36)$	$R/((0.8*10^{(-L*.00003)})+1.29)$	$R/((0.2*10^{(-L*.00003)})+1.56)$
4	R/(1.45)	R/(1.51)	$R/((0.7*10^{(-L*.00002)})+1.46)$	$R/((.07*10^{(-L*.00002)})+1.85)$
ALL	R/(1.72)	R/(1.72)	R/(1.96)	R/(2.27)

R = the instantaneous data rate
L = the number of bits to be transmitted

J.3.4.9.1 Terminals lacking hop detection. The ALL case in Table J-3 is to show the efficiency of the multi-dwell protocol in systems where the hop cannot be detected due to hardware or software limitations. Since there is no hop timing information available, the DTE shall assume that the radio will hop at every possible time slot. In these systems, it is assumed that timing synchronization with the radio will be made by the detection of the falling edge of the radio delayed push to talk (DPTT) signal provided by the HAVEQUICK II compatible radio.

J.3.5 Robust Communications Protocol Network Timing. The use of the robust communications protocol requires modification to some of the Appendix C type 1 network timing equations. The bit rate, transmit delays, and receive processing delays are modified by the robust protocol. For purposes of robust network timing, two system bit rates are defined. The first is the channel bit rate which is represented as n_c. The second is the data link bit rate which is represented as n_l. As an example, if rate 1/3 convolutional coding is applied at the physical layer and the channel bit rate is 16 kbps, the link bit rate would be 5.33 kbps. In this example, an external cryptographic device would transmit the MI field at n_c Hz and an internal cryptographic device would transmit the MI field at n_l Hz. The multi-dwell reduction of n_l is not deterministic but is bounded. The average multi-dwell n_l is a function of the multi-dwell packet format, the timing of the DTE transmit request in relation to the radio TRANSEC timing, and the number of bits to be transmitted. The following Type 1 network access control subfunctions are specified in Appendix C:

a. network busy sensing

b. response hold delay (RHD)

c. timeout period (TP)

d. network access delay (NAD)

The following subparagraphs address required modifications to network timing equations associated with these subfunctions as a result of using the robust communications protocol.

J.3.5.1 Net busy sensing. Because net busy sensing is performed at the physical level, there are no modifications to the net busy sensing timing or methods when using the robust communications protocol.

J.3.5.2 Response hold delay. The additional transmission time required for the robust synchronization field and n_l bit rate reductions impact the response transmission time parameter, (S), contained in the response hold delay timing equation, RHD_0. Also additional receive processing delays impact the internal DTE timing calculations. The RHD_0 is calculated as follows:

$$RHD_O = EPRE + PHASING + S + ELAG + TURN + TOL$$

J.3.5.2.1 Response transmission time (S). The response transmission time is changed by the robust protocol. A Type 1 Response PDU from the data link layer consists of the 64-bit message synchronization field, the 75-bit robust frame format, an optional embedded COMSEC MI field, the 168-bit word count and Transmission Header TDC block, and 384, 168, or 80 bits of acknowledgment data depending on the selectable use of EDC and TDC. The 64-bit message synchronization field and the 75-bit robust frame format are transmitted at the channel bit rate (n_c) The remaining components are transmitted at the link data rate (n_l).

J.3.5.2.1.1 Multi-dwell Response. Where multi-dwell is used to send the original message at a channel bit rate, n_c, of 16 Kbps, all responses, i.e. Type 1 acknowledgments, except for a secure external crypto transmission, are short enough that a multi-dwell transmission is not required. A multi-dwell transmission is required when using an external crypto because the data may be interrupted by a frequency hop. Table J-4 gives the maximum number of bits that will be transmitted at the channel bit rate (n_c) for the link data sizes and multi-dwell SOP majority logic choices. These numbers are for the HOP ALL case, which is the worst case, and for the highest operational hop rate, hop rate 3. The 139 robust protocol header bits are included in Table J-4. The numbers in Table J-4 do not include the “wasted time” shown in Figure J-7.

TABLE J-4. Multi-dwell external crypto response transmission time
HRT= 15.6 ms, mode 3, (MV3:5, 6 segments/packet)

LINK BITS	FEC 1/3 Hop Rate 3	FEC 1/3 Hop All	No EC Hop Rate 3	No FEC Hop All
312	$2139/n_c$	$3139/n_c$	$662/n_c$	$662/n_c$
392	$2139/n_c$	$4143/n_c$	$1185/n_c$	$1185/n_c$
616	$3139/n_c$	$5189/n_c$	$1185/n_c$	$1708/n_c$

NOTES: Table J-4 is optimized for the AN/ARC-164.
Robust Mode 3 is used for response PDUs. The choices are FEC or no FEC.
LINK BITS = the number of bits sent to the physical layer in a response PDU.
If Delay PTT is not supported by all radios in the network, then only columns marked "Hop All" apply.
Columns marked "Hop All" are required. Columns marked "Hope Rate 3" are optional.

J.3.5.2.1.2 Non Multi-dwell Response. Where an external crypto device is not used and n_c is 16 Kbps, the long dwell time will contain the entire response. Where an external crypto device is used, convolutional encoding is not used and the n_c is 16 Kbps, the crypto preamble will be contained within the long dwell time and the response will be contained within the first minimum dwell period following the long dwell time.

J.3.5.2.2 Response transmission example. Figure J-7 shows an example of the timing of an acknowledgment when an external cryptographic device is used with the HAVEQUICK II radio. The falling edge of the DPTT signal marks the beginning of a long hop dwell that is long enough to contain the crypto preamble time.

If an external crypto device was not used, this long dwell time would contain the entire acknowledgment. After the crypto has finished transmitting the MI field, the transmitting DTE begins to supply data for transmission. Typically, the COMSEC bit synchronization time is not very accurate and may be long enough to push the MI field to the end of the guaranteed long dwell time. For this reason, the DTE shall wait to start data transmission on the first hop dwell following the long guaranteed dwell. The end of the guaranteed hop dwell is marked by the possible hop label. The first bit of the robust SOM pattern is transmitted after the configured hop recovery time (HRT). During the transmission of the response, one or more hops may occur which will vary the transmission time of the acknowledgment. When the response transmission is complete, the DTE de-asserts the transmit request signal. The radio will de-assert DPTT after a variable delay (ET_1) at a time synchronized with the hop sequence. After DPTT is de-asserted, the radio RF output remains active and a radio hop will not occur. This allows for the transmission of the crypto postamble. The radio RF output remains active for longer than is required for the transmission of the crypto postamble, which is shown as the ET_2 time period in Figure J-7. For HAVEQUICK II radios, ET_1, crypto postamble PLUS ET_2 equals the transmitter turnaround time, $(TTURN = ET_1 + ET_2)$, as defined in Appendix C.

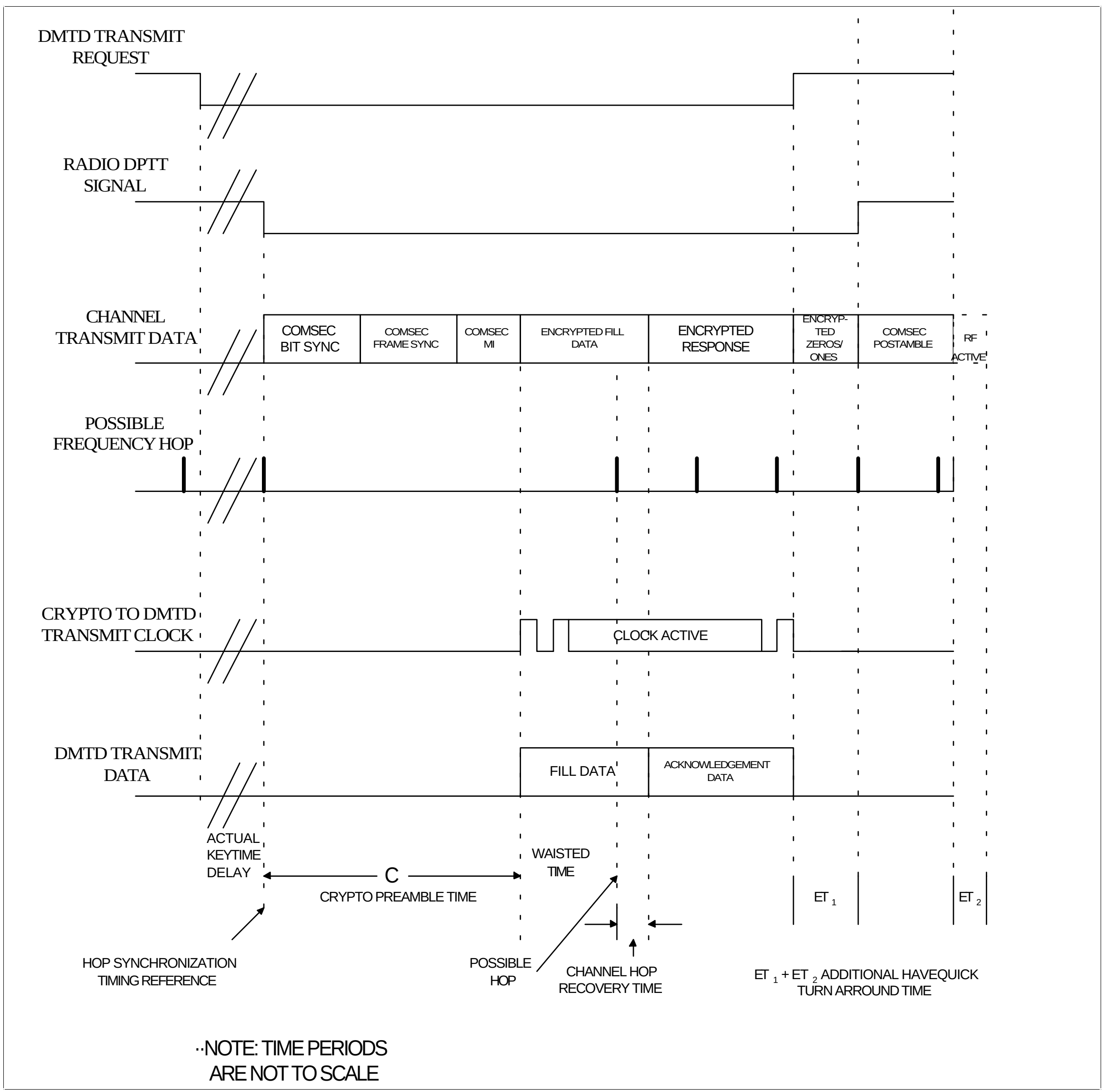


FIGURE J-7. HAVEQUICK II external crypto acknowledgment transmission.

J.3.5.2.3 Estimation of multi-dwell n_l. Figure J-8 shows an example of the n_l data rate for a multi-dwell transmission with a channel data rate of 16 kbps. This is the worst case data rate reduction which would be experienced with rate 1/3 convolutional coding, a 64 bit SOP pattern length, and 3 out of 5 majority logic decoding of the segment count field. The data rate shown Figure J-8 is the number of link bits to transmit divided by the number of channel bits transmitted times the n_c. Since rate 1/3 convolutional encoding is used in this example, the maximum link data rate achievable would be 5.33 kbps. For short messages, the radio hop timing at the beginning of the transmission has a significant impact on the transmission efficiency. This example uses 13 segments per packet which is the recommended segment per packet count for long transmissions using 3 out of 5 majority logic. This figure and the equations given in Table J-3 are given as an aid for network throughput estimation and should not be used for network timing. The bit rate estimating equation used in Figure J-3 is:

$$\text{link rate} = nc / (0.5*10^{(\text{-link bits} * .00003)} + 1.301)$$

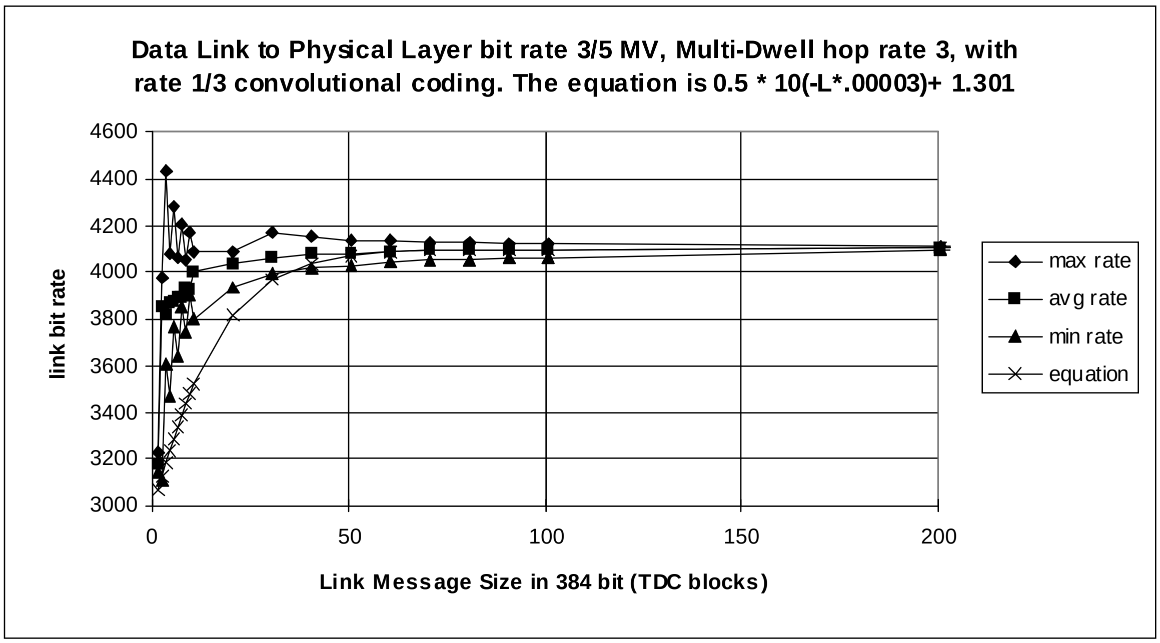


FIGURE J-8. Link Data Rate as a Function of Message Size.

J.3.5.2.4 Receive Processing Delays. In order to calculate the reference point for the RHD and TP timers, the receiving DTE must know the time of arrival of the last bit of the transmission. In order to do this, the data link layer normally determines the last bit of the transmission after decoding the word count and tags the arrival of the last data bit from the physical layer. The physical layer receive delays are dependent on the DTE hardware and software implementation. The two delay components are processing delays and data pipeline delays. The processing delays are independent of the received data rate and the pipeline delays are dependent on the data rate. If the receive data rate is known, the data link layer can calculate the time of arrival of the last bit of the message by subtracting off the processing and pipeline delays. If the received data rate is not known, it is impossible to convert a pipeline delay from bits to seconds. The data rate of all non-multi-dwell transmissions is known to be either n_c or $n_c/3$ dependent on the use of rate 1/3 convolutional coding. The received data rate of a multi-dwell transmission is not known. For this reason, when a multi-dwell transmission is received, the physical layer must tag the time of arrival of the final multi-dwell bit. The physical layer can determine the time of arrival of the last bit by using the end of frame flag which is the most significant bit in the final multi-dwell segment count field. A logical signal from the physical layer to the data link layer indicating the message completion time is required to insure that the transmitter and receiver(s) use the same reference point for the calculation of RHD and TP.

The trace back buffer length of the Viterbi decoder introduces a known pipeline delay in the received data. The length of the trace back buffer is an implementation choice which is dependent on the Viterbi decoder architecture. Pipeline delay is the time needed to flush the trace back buffer.

J.3.5.3 Timeout period (TP). The timeout period is derived from the following equations as described in Appendix C:

$$TP = (j \times RHD_0) + \text{Maximum}(DTEACK, TURN)$$
$$TP = \text{Maximum}(DTEPROC, TURN)$$
$$TP = (15 \times RHD_0) + TOL + TURN$$

Modifications to the timeout period are result of changes to RHD_0 and DTE receive processing delays, which have been addressed in paragraphs J.3.5.2 and its subparagraphs.

J.3.5.4 Network access delay (NAD). There are no modifications to the network timing equations associated network access delay. The network access delay is always an integer number times the Net_Busy_Detect_Time which, as previously discussed, has not been modified.

J.3.6 Application guidance for the HAVEQUICK II link.

J.3.6.1 Frequency Hop Synchronization. The HAVEQUICK II TRANSEC timing and the DTE network timing are not synchronized. To avoid the loss of critical data, such as the cryptographic synchronization and/or the protocol SOM patterns, the DTE transmission timing must be synchronized to the frequency hops. The radio should provide a DPTT signal which marks the beginning of a hop dwell with a guaranteed minimum duration. This minimum dwell period is sufficient to carry the synchronization field of an external cryptographic device or the robust frame synchronization field when an internal cryptographic device is used.

J.3.7 Summary. The physical layer robust protocol introduces additional transmit and receive delays due to the robust header and the convolutional decoding pipeline delays. Multi-dwell packetizing introduces a data rate reduction which varies widely for short transmissions. The

HAVEQUICK II radio introduces variable delays in the keytime delay and the equipment turn-around time. To maintain network timing using the type 1 timing equations, the net busy sense timing and the response transmission time must be a known constant. In most cases, the response can be transmitted without using the multi-dwell packetizing algorithm. When the multi-dwell packetizing algorithm must be used to transmit a response, the worst case time to complete the transmission is used in the response transmission time component of the term TURN in RHD. The message transmission time is variable and is only required to be known at the end of the transmission. Two additional physical to data link signals are required to mark the time of the last transmitted bit for transmission, and the time of the last received bit for a reception.

APPENDIX K

BOSE - CHAUDHARI - HOCQUENGHEM (15, 7) CODING ALGORITHM

K.1. General.

K.1.1 Scope. This appendix describes a linear block cyclic code capable of correcting any combination of two or fewer errors in a block of 15 bits.

K.1.2 Application. This appendix is a conditionally mandatory part of MIL-STD-188-220. It is mandatory for implementing the Robust Communications Protocol described in Appendix J.

K.2. Applicable documents. This section is not applicable to this appendix.

K.3. BCH (15,7) code. The BCH (15,7) code is a linear, block, cyclic, BCH code capable of correcting any combination of two or fewer errors in a block of 15 bits. The generator polynomial for this code is

$$g(x) = 1 + X^4 + X^6 + X^7 + X^8$$

where g(x) is a factor of $X^{15} + 1$

K.3.1 Hardware encoding. BCH (15, 7) encoding can be performed with an 8 stage feedback shift register with feedback connections selected according to the coefficients of g(x). A shift register corresponding to the coefficients of g(x) is shown in Figure K-1.

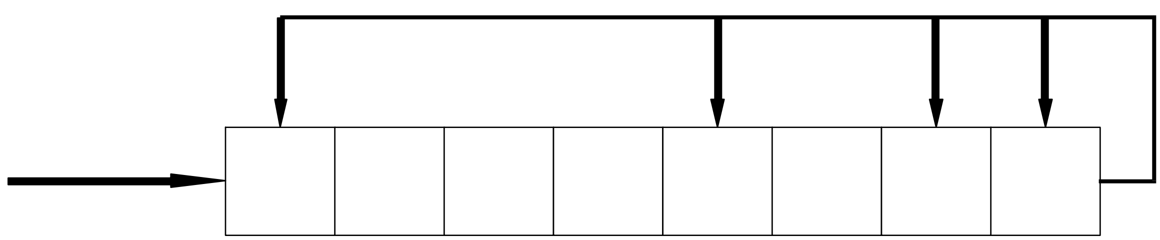
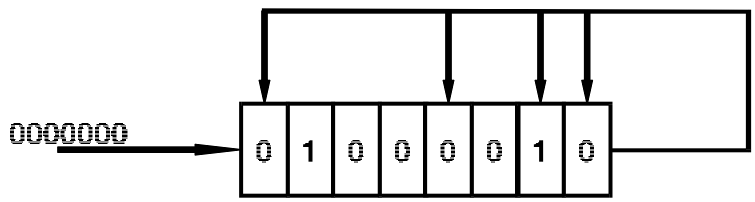


FIGURE K-1. Shift register encoder for the BCH (15, 7) code.

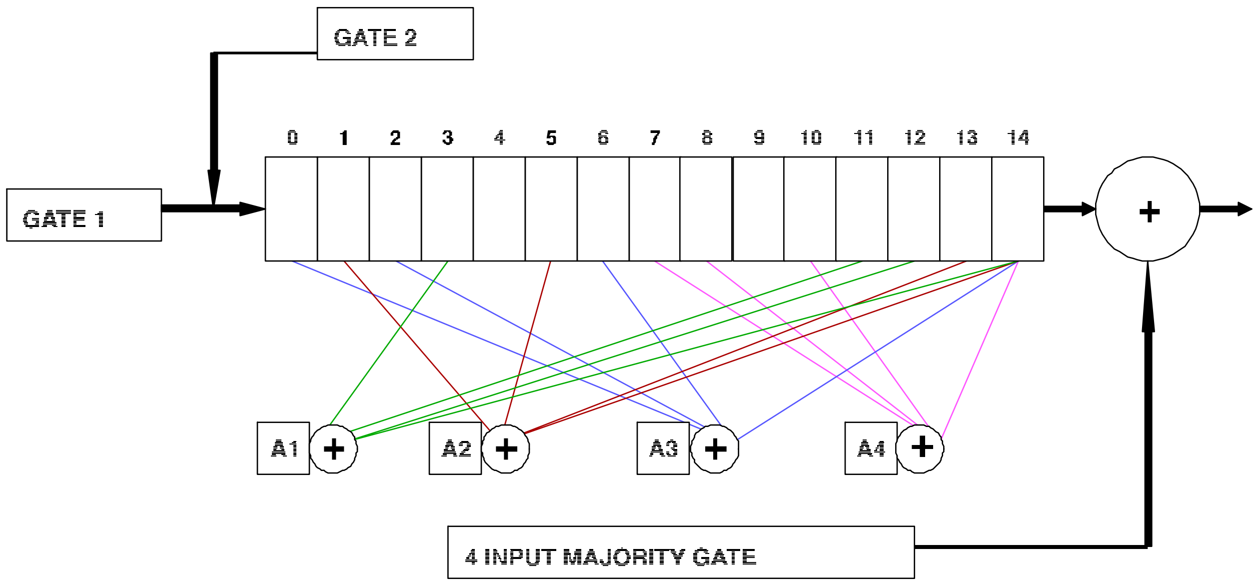
Figure K-2 illustrates its operation by showing the encoding of the information vector (1000010) to form the code vector (10100101 | 01000010), where the parity check sequence is shown before the partition and the information sequence after. The information sequence with eight zeros after it (place holders for the parity bits to be calculated) is shifted into the register initially (it is really a fifteen bit shift register but only the last eight positions correspond to the coefficients of g(x) and contain feedback connections). The operation of the shift register consists of seven rounds of shift, feedback, and sum operations. The parity portion of the code vector can then be read out of the shift register as shown.



Shift	000000	0	0	1	0	0	0	0	1	0
Feedback		0				0		0	0	
Sum	000000	0	0	1	0	0	0	0	1	
Shift	00000	0	0	0	1	0	0	0	0	1
Feedback		1				1		1	1	
Sum	00000	1	0	0	1	1	0	1	1	
Shift	0000	0	1	0	0	1	1	0	1	1
Feedback		1				1		1	1	
Sum	0000	1	1	0	0	0	1	1	0	
Shift	000	0	1	1	0	0	0	1	1	0
Feedback		0				0		0	0	
Sum	000	0	1	1	0	0	0	1	1	
Shift	00	0	0	1	1	0	0	0	1	1
Feedback		1				1		1	1	
Sum	00	1	0	1	1	0	0	0	1	
Shift	0	0	1	0	1	1	1	0	1	0
Feedback		0				0		0	0	
Sum	0	0	1	0	1	1	1	0	1	
Shift		0	0	1	0	1	1	1	0	1
Feedback		1				1		1	1	
Sum		1	0	1	0	0	1	0	1	

FIGURE K-2. Encoding example.

K.3.2 Hardware/Software Decoding. Because of its special structure (it is completely orthogonalizable in one step), the BCH (15,7) code can be decoded very efficiently with a majority logic scheme which can be directly implemented in software or hardware. It is most easily described in terms of the shift register implementation shown in Figure K-3. With gate 2 open and gate 1 closed, the received block is read into the shift register. the output of the four modulo 2 summers is sampled by the majority gate and processed as follows: if a clear majority of the inputs are ones (three or more) then the output is one, otherwise (if two or fewer inputs are ones) the output is zero. This output is used to correct the last bit of the shift register. The corrected bit is output to the receiver and feedback through gate 2 as the register is right shifted. he process is now repeated thirteen times until the last bit is corrected.



A1	3	11	12	14
A2	1	5	13	14
A3	0	2	6	14
A4	7	8	10	14

FIGURE K-3. BCH (15, 7) majority logic decoding.

K.3.3 Software encoding. The BCH (15,7) code is most efficiently encoded in systematic form from the generator matrix shown in Figure K-4.

$$\mathbf{G} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}}_{\textbf{Parity}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\textbf{Identity}}$$

FIGURE K-4. BCH (15, 7) generator matrix.

CONCLUDING MATERIAL

Custodians:

Army - CR
Navy - EC
Air Force - 90

Preparing Activity:

DISA - DC

Review activities:

Army - AM, SC, PT
Navy - MC, NC, TD
Air Force - 02, 13, 17, 19, 29, 89, 90, 93
DLA - DH
NSA - NS
ECAC - --
DMA - MP
DOT - OST
DIA - DIA

Agent:

N/A

(Project TCSS - 2200)

User Activities:

Army - AC, MI
Navy - OM
Air Force - 11, 18

International Interest:

NATO

STANDARDIZATION DOCUMENT IMPROVEMENT PROPOSAL

INSTRUCTIONS

1. The preparing activity must complete blocks 1,2, 3, and 8. In block 1, both the document number and revision letter should be given.
2. The submitter of this form must complete blocks 4, 5, 6, and 7.
3. The preparing activity must provide a reply within 30 days from receipt of the form.

NOTE: This form may not be used to request copies of documents, nor to request waivers, or clarification of requirements on current contracts. Comments submitted on this form do not constitute or imply authorization to waive any portion of the referenced document(s) or to amend contractual requirements.

I RECOMMEND A CHANGE:	1. DOCUMENT NUMBER MIL-STD-188-220B	2. DOCUMENT DATE *(YYMMDD)* 980120

3. DOCUMENT TITLE Interoperability Standard for Digital Message Transfer Device Subsystems

4. NATURE OF CHANGE *(Identify paragraph number and include proposed rewrite, if possible. Attach extra sheets as needed.)*

5. REASON FOR RECOMMENDATION

6. SUBMITTER

a. NAME *(Last, First, Middle Initial)*	b. ORGANIZATION	
c. ADDRESS *(Include Zip Code)*	d. TELEPHONE *(Include Area Code)* (1) Commercial (2) DSN *(If applicable)*	7. DATE SUBMITTED *(YYMMDD)*

8. PREPARING ACTIVITY **DEFENSE INFORMATION SYSTEMS AGENCY (DISA)**

a. NAME MR. GERALD RING	b. TELEPHONE *(Include Area Code)* (1) Commercial 732-427-6893 (2) DSN 987-6893
c. ADDRESS *(Include Zip Code)* DIR, JOINT INTEROPERABILITY ENGINEERING BUILDING 283 ATTN: JIEO-JEBB (G. Ring) FT. MONMOUTH, NJ 07703	**IF YOU DO NOT RECEIVE A REPLY WITHIN 45 DAYS, CONTACT:** Defense Quality and Standardization Office 5203 Leesburg Pike, Suite 1403, Falls Church, VA 22041-3466 Telephone (703) 756-2340 DSN 289-2340